

**VISUALISASI SEBARAN ALUMNI UNIVERSITAS ISLAM
NEGERI SULTAN SYARIF KASIM RIAU
(STUDI KASUS FAKULTAS SAINS DAN TEKNOLOGI)**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

oleh :

BOBBY HANDOKO MASRI

10651004329



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU**

2013

VISUALISASI SEBARAN ALUMNI UNIVERSITAS ISLAM NEGRI SULTAN SYARIF KASIM RIAU

(STUDI KASUS: Fakultas Sains dan Teknologi)

BOBBY HANDOKO MASRI
10651004329

Tanggal Sidang: 20 Juni 2013
Periode Wisuda: November 2013

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

Alumni UIN SUSKA RIAU telah tersebar di berbagai penjuru wilayah di Indonesia, pendataan sebaran alumni pada umumnya dilakukan dengan memanfaatkan Sistem Informasi yang menyajikan data tekstual yang diambil dari sebuah *database*. Untuk memenuhi kebutuhan visualisasi, digunakan beberapa kolaborasi yaitu Google Map sebagai peta geospasial yang menampilkan tampak sebaran alumni UIN SUSKA RIAU secara geografis. Widget Timeline yang digunakan untuk melihat jejak waktu keberadaan alumni dalam *time scroll* untuk setiap *event*, dan sebuah *database* sebagai media penyimpanan data baik text atau gambar, informasi sebaran alumni tampil berdasarkan *data filter* dengan kriteria tertentu sesuai kebutuhan *user*, alumni mendapatkan akses yang didaftarkan oleh jurusan masing-masing alumni. Hasil dari pembangunan ini memberikan manfaat bagi jurusan dalam melacak keberadaan alumni dengan adanya visualisasi sebaran yang juga memberikan laporan keberadaan alumni.

Kata Kunci : Alumni, *Peta Geospasial, Timeline, Google Map*

VISUALIZATION DISTRIBUTION OF ALUMNI ISLAMIC UNIVERSITY NEGRI SULTAN SYARIF KASIM RIAU

(CASE STUDY: Faculty of Science and Technology)

BOBBY HANDOKO MASRI
10651004329

Session Date: June 20, 2013
Graduation period: November ,2013

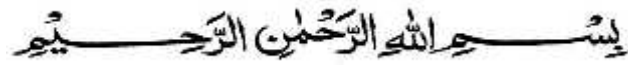
Department of Informatics
Faculty of Science and Technology
State Islamic University of Sultan Syarif Kasim Riau

ABSTRACT

Alumni UIN Suska RIAU been scattered across the regions in Indonesia, the distribution of the alumni data collection is generally done by using System Opera-which presents textual data retrieved from a database. To meet the needs of visualization, collaboration is used several Google Map as a geospatial map showing the distribution of alumni seem UIN Suska RIAU geographically. Timeline widget is used to see where the time traces of alumni scroll in time for each event, and a database as a data storage medium, either text or images, information distribution based on the data of alumni perform certain filter criteria in accordance with user requirements, graduates gain access registered by the department each alumnus. Result of this development to benefit the department in keeping track of where the alumni with the visualization of the distribution which also reports the presence of alumni giving.

Keywords: Alumni, Geospatial Map, Timeline, Google Map

KATA PENGANTAR



Assalamu'alaikum Wr. Wb.

Alhamdulillahirabbil'alamin segala puji bagi ALLAH SWT yang telah melimpahkan rahmat dan hidayah_Nya, sehingga penulis dapat menyelesaikan skripsi ini. Sholawat beriring salam kita hadiahkan kepada baginda Rasulullah Saw, buah hati Aminah, Kekasih ALLAH SWT, teladan bagi seluruh umat dengan mengucapkan "*ALLAHUMMA SHOLLI'ALA MUHAMMAD WA'ALA ALI MUHAMMAD*" .

Tugas Akhir dengan judul " **VISUALISASI SEBARAN ALUMNI UNIVERSITAS ISLAM NEGRI SULTAN SYARIF KASIM RIAU** (Studi kasus: Fakultas Sains Dan Teknolgi)" ini disusun sebagai syarat kelulusan dalam menyelesaikan studi di Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau.

Banyak sekali pihak yang telah membantu penulis dalam melaksanakan dan menyusun laporan ini, baik berupa materi maupun berupa moril atau motivasi. Untuk itu pada kesempatan ini, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. H. M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Bapak Dra. Hj. Yenita Morena, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau.
3. Ibu Dr. Okfalisa, S.T, M.Sc selaku ketua jurusan Teknik Informatika Fakultas Sains dan Teknologi yang dalam hal ini banyak memberikan bantuan dan dukungan dalam proses penyelesaian tugas akhir ini.
4. Bapak Benny Sukma Negara, S.T, M.T, selaku pembimbing I yang selalu memberikan bimbingan dan petunjuk sehingga tugas akhir ini dapat diselesaikan dengan baik.

5. Bapak M. Irsyad, M.T selaku penguji I dan M. Syafrizal, S.T. M.Cs selaku penguji II yang telah memberikan masukan yang bermanfaat kepada penulis.
6. Bapak Reski Mai Candra, S.T. M.Sc, selaku koordinator tugas akhir jurusan Teknik Informatika.
7. Keluarga saya Ayahanda Masri. B dan Ibunda Srimariani terima kasih banyak atas pengorbanannya memberikan dukungan dan semangat demi kesuksesan dan kejayaan anak-anaknya. Dan selalu menjadi inspirasi, motivasi hidupku dalam setiap langkahku untuk menyelesaikan tugas akhir ini.
8. Adekku Noufi Saputra, Amd, Imam Wicaksono, Teguh Prasetyo, dan annisya yang selalu memberikan semangat dan motivasi untuk menjadi lebih baik.
9. dr. Raudhatul Jannah yang telah memberikan dorongan dan dukungan untuk selalu menyelesaikan tugas akhir ini.
10. Segenap dosen Teknik Informatika yang tidak dapat saya sebutkan satu persatu yang telah banyak memberikan ilmu dan bimbingan akademis kepada penulis selama masa perkuliahan.
11. Novriady, Fristian, Devi, Rian, Arel, Andika, Try, Pipit, Rafika, Reni, Melya, Eka, Jason, Bayu, Aang, Ade dan Merry terima kasih atas bantuan dan motivasi serta kebersamaan yang telah diberikan selama ini.
12. Teman seperjuangan yang telah membantu dan memotivasi penulis dalam penyelesaian laporan tugas akhir ini, khususnya COMMIT angkatan 2006.

Dalam penulisan laporan ini, penulis menyadari bahwa masih banyak kekurangan dan harus diperbaiki. Untuk itu penulis membuka diri dalam menerima masukan berupa kritik dan saran yang membangun dari semua pihak untuk penyempurnaan laporan ini dan lebih baik di masa yang akan datang.

Dan akhir kata penulis berharap semoga laporan tugas akhir ini dapat bermanfaat bagi semua pihak.

Pekanbaru, Juni 2013

Penulis

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	iv
LEMBAR PERNYATAAN	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xvi
BAB I PENDAHULUAN	
1.1.Latar Belakang	I-1
1.2.Rumusan Masalah	I-2
1.3.Batasan Masalah.....	I-2
1.4.Tujuan Penelitian.....	I-3
1.5.Sistematika Penulisan	I-3
BAB II LANDASAN TEORI	
2.1. <i>Simile Timeline</i>	II-1
2.1.1.Tampilan <i>Widget Timeline</i>	II-1
2.1.2.Format Data <i>Attribute Event</i>	II-2
2.2. <i>PHP (Hypertext Preprocessor)</i>	II-3
2.3. <i>MySQL</i>	II-4
2.4. <i>Google MAP</i>	II-4
2.4.1.Tampilan <i>Widget Timeline</i>	II-5
BAB III METODOLOGI PENELITIAN	
3.1.Identifikasi Masalah	III-2
3.2.Perumusan Masalah	III-2
3.3.Pengumpulan Data	III-2

3.3.1.Studi Pustaka	III-2
3.4.Analisa Sistem.....	III-3
3.4.1.Analisa Penelitian Sebelumnya.....	III-2
3.4.1.Analisa Sistem Baru.....	III-5
3.4.2.1.Subsistem Data.....	III-5
3.4.2.2.Subsistem Model.....	III-5
3.4.2.3.Subsistem Dialog	III-5
3.5.Perancangan	III-5
3.5.1.Subsistem Data.....	III-5
3.5.2.Subsistem Model.....	III-6
3.5.3.Subsistem Dialog	III-6
3.6.Implementasi	III-6
3.7.Pengujian.....	III-6
3.8.Kesimpulan dan Saran.....	III-6

BAB IV ANALISA DAN PERANCANGAN

4.1.Desain Arsitektur Sistem	IV-1
4.2.Context Diagram.....	IV-1
4.3.Data Flow Diagram	IV-2
4.3.1.DFD Level 2.1	IV-4
4.3.2.DFD Level 2.2	IV-4
4.3.3.DFD Level 2.3	IV-5
4.3.4.DFD Level 2.4	IV-5
4.4.Entity Relationship Diagram	IV-6
4.5.Flowchart Sistem.....	IV-10
4.6.Site Flow Diagram.....	IV-17

BAB V IMPLEMENTASI DAN PENGUJIAN

5.1.Implementasi	V-1
5.1.1.Kebutuhan Sistem	V-1
5.2.Koneksi Database	V-2
5.3.Service Google Map	V-3
5.4.Service SIMILE Timeline	V-7

5.5.1. <i>User Administrator</i>	V-11
5.5.2. <i>User Alumni</i>	V-14
BAB VI PENUTUP	
6.1. Kesimpulan	VI-1
6.2. Saran.....	VI-1
DAFTAR PUSTAKA	xvii
DAFTAR RIWAYAT HIDUP.....	xviii
LAMPIRAN	

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sistem Visualisasi Peta merupakan aplikasi berbasis *WebGIS* yang mampu menyajikan data *geo-spatial* secara kartografis. Sistem visualisasi peta ini dikembangkan dengan menggunakan standar *Open Geospatial Consortium* (OGC) dengan menggunakan *Web Map Service* (WMS) dan *Web Feature Service* (WFS). Dengan layanan WMS dan WFS ini memungkinkan pengguna melakukan akses data spasial baik melalui aplikasi *desktop* maupun aplikasi berbasis *web*. Saat ini banyak sekali dikembangkan berbagai aplikasi berbasis peta untuk berbagai keperluan. Aplikasi ini dapat digunakan untuk berbagai macam keperluan yang berkaitan dengan data spasial (peta), seperti sistem informasi inventaris, sistem informasi potensi investasi daerah, sistem informasi potensi pariwisata, dan sistem informasi potensi daerah.

Setiap tahunnya UIN SUSKA meluluskan hingga ratusan mahasiswa dari semua jurusan. Alumni merupakan parameter penting bagi UIN SUSKA untuk melakukan evaluasi melalui pendataan untuk melihat sinergi antara pendidikan yang diberikan, hasil didikan, dan kecocokannya dengan kebutuhan industri akan sumber daya manusia yang berkualitas sebagai kepentingan akreditasi. Saat ini pendataan alumni masih dilakukan dengan pencatatan yang disimpan dalam dokumen *office*, untuk mendapatkan informasi tersebut pihak kampus yang membutuhkan harus meminta dokumen alumni tersebut secara langsung kepada *staf* jurusan yang mengumpulkan informasi alumni. Data yang disajikan dalam bentuk angka atau *table*, relatif sulit untuk dipahami sehingga dibutuhkan model informasi berbasis visualisasi.

Adapun penelitian yang terkait visualisasi sebelumnya di lakukan oleh Setiani, Teknik Informatika dengan judul penelitian “*Visualisasi Peta Menggunakan Spasial Database pada Sistem Informasi Kepariwisata*an”. Pada penelitian tersebut, aplikasi memvisualisasikan peta kedalam perangkat bergerak sebagai *client* berdasarkan data *vector* yang berada pada *server*. Aplikasi tersebut dibangun

menggunakan visual studio 2008 dan *SQL server* 2008 sebagai *server*. Komunikasi antara aplikasi klien di perangkat bergerak dengan *server* menggunakan *web service*. Namun penelitian diatas memiliki kelemahan karena pengembang aplikasi tidak mudah mengembangkan program sesuai kebutuhan. Hal ini disebabkan oleh struktur data *vector* yang lebih kompleks dan prosedur fungsi dan analisisnya memerlukan kemampuan tinggi karena lebih sulit. Pengguna harus membeli *system* perangkat lunaknya karena teknologinya masih mahal. Prosedurnya terkadang lebih sulit. Karena proses keseluruhan untuk mendapatkannya lebih lama, peta *vector* seringkali mengalami *out of date* atau kadaluarsa. Serta memerlukan perangkat keras dan perangkat lunak yang lebih mahal untuk mengolah data *vector*. Kemudian data *vector* memiliki *overlay* beberapa *layers vector* secara simultan memerlukan waktu yang relative lama.

Adapun pada penelitian yang akan penulis lakukan tidak menggunakan data *vector* seperti yang digunakan oleh peneliti sebelumnya. Penulis dalam menampilkan peta menggunakan fasilitas yang telah disediakan oleh *google Map*, penulis menggunakan fasilitas *google Map* mengingat pada saat ini peta dari *google Map* tidak memerlukan spesifikasi *hardware* yang tinggi untuk mengolah peta yang disediakan *google Map*. Pada penelitian yang akan penulis bangun adalah menampilkan posisi kerja terbaru dari masing-masing alumni di peta. Pengguna *web* dapat melihat biodata alumni dan periode masa kerja dengan meng-klik *point* di peta yang disajikan.

1.2 Rumusan Masalah

Masalah yang dihadapi dalam membuat Tugas Akhir ini adalah bagaimana membuat visualisasi sebaran alumni disajikan dalam sistem informasi berbasis *web*.

1.3 Batasan Masalah

Batasan masalah dalam visualisasi sebaran alumni UIN SUSKA adalah:

1. Sistem Informasi menyajikan informasi sebaran alumni secara visual dengan menggunakan peta *geospasial* dan *timeline*.

2. Peta *geospasial* mewakili sebaran alumni UIN SUSKA yang berbasis lokasi menggunakan *Google Map API*,
3. *Timeline* yang mewakili sebaran alumni UIN SUSKA yang berbasis waktu kerja di suatu perusahaan menggunakan *SIMILE* .

1.4 Tujuan Penelitian

Tujuan pembuatan proyek akhir ini adalah:

1. Alumni dapat memberikan informasi tempat mereka bekerja dan awal waktu mereka mulai bekerja pada tempat tersebut.
2. Jurusan dapat mencari informasi alumni dan langsung melihat keberadaan alumni dengan menggunakan *Google Map*.

1.5 Sistematika Penulisan

Sistematika penulisan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, serta sistematika penulisan laporan tugas akhir.

BAB II LANDASAN TEORI

Bab ini membahas teori-teori yang berhubungan dengan topik penelitian, yang terdiri dari Visualisasi Sebaran Alumni UIN SUSKA Berbasis *Website*.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang metodologi yang digunakan dalam penelitian dan pengembangan perangkat lunak.

BAB IV ANALISA DAN PERANCANGAN

Bab ini membahas tentang hasil analisa, deskripsi sistem, karakteristik pengguna, deskripsi umum kebutuhan, deskripsi perancangan rinci dan perancangan antar muka sistem

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas implementasi dan pengujian yang dilakukan terhadap Visualisasi Sebaran Alumni UIN SUSKA Berbasis *Website*.

BAB VI PENUTUP

Bab ini berisi kesimpulan yang dihasilkan dari pembahasan tentang Visualisasi Sebaran Alumni UIN SUSKA Berbasis *Website*. Dan beberapa saran sebagai hasil akhir dari penelitian yang telah dilakukan.

BAB II

LANDASAN TEORI

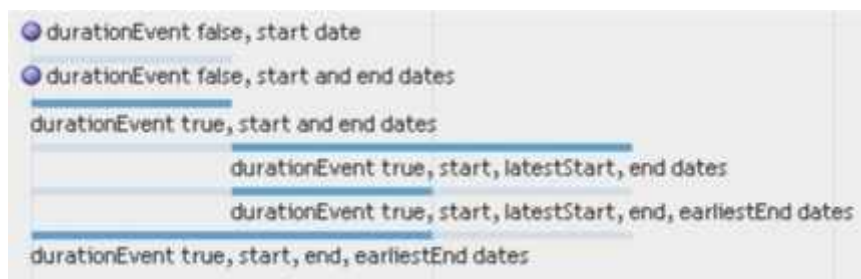
2.1 *SIMILE Timeline*

SIMILE yang merupakan akronim dari “*Semantic Interoperability of Metadata and Information in unLike Environments*” adalah sebuah proyek yang dikerjakan oleh *Massachusetts Institute of Technology (MIT)*. *SIMILE* terdiri dari beberapa proyek *open source* yang menunjukkan *tools* dalam konteks teknologi-teknologi *semantic-web-enabling*. (*SIMILE Project*, 2008 dan *STEFANO*, 2007).

Salah satu proyek *SIMILE* adalah *timeline* berupa *widget* interaktif yang dapat memvisualisasikan informasi berbasis waktu. *Timeline* memungkinkan *user* melakukan penyeretan interaktif dengan *pop-up* keterangan, satu atau dua *time band* yang bergerak, beserta *link* dan gambar

2.1.1 *Tampilan Widget Timeline*

Dalam penggunaan standar, ada beberapa format dalam menampilkan *event* dalam *timeline*, variasi format tersebut dapat dilihat pada gambar 2.1:



Gambar 2.1 Variasi format *event* dalam *timeline*.

Sumber : <http://simile-widgets.org/wiki/Image:Timeline-event-displays.jpg>

Dari contoh variasi *timeline* tersebut, dapat dilihat perbedaan masing-masing bentuk *timeline* dari panjang, *icon*, dan warna, terdapat 2 jenis *event* dalam *timeline* dengan penjelasan sebagai berikut

1. *Non-Duration Events (Instant Events)*

Non-Duration event, juga dikenal dengan *instant event*, *event* ini fokus pada waktu tertentu. Salah satu contoh dalam sejarah seseorang berupa tanggal wafat yang hanya terjadi pada satu waktu.



Gambar 2.2 *Instant event*

2. *Duration Events.*

Duration event, *event* yang terjadi selama periode waktu. Contoh dalam sejarah adalah tanggal perang. *Duration event* dapat dilihat pita polos tanpa *icon*. Perhatikan bagaimana *event* label sejajar dengan awal pita polos.



Gambar 2.3 *Duration event* dengan periode pasti.

Sumber : <http://simile-widgets.org/wiki/Image:Timeline-event-displays.jpg>

2.1.2 **Format Data Attribute Event**

Data untuk *Timeline Event* bisa menggunakan salah satu dari jenis format berikut:

1. *XML*
2. *JSON*
3. *SPARQL*

Hanya ada satu maksud yang mendasari atribut *event Timeline*. Tapi atribut *event* disusun sedikit berbeda dalam format *transport* data yang berbeda. Setiap adapter format membaca data yang masuk dan menyimpannya dalam *Timeline*. Sebagai contoh, dalam format *XML*, *event* adalah serangkaian elemen di bawah elemen *root*. Dalam *JSON*, hash *root* memiliki sebuah *event array*, yang urutannya tidak penting.

Berikut adalah atribut-atribut dasar *timeline* adalah sebagai berikut:

1. ***start***, format tanggal (contoh "*May 20 1961 00:00:00 GM- 0600*")
2. ***latestStart***, untuk awal waktu yang tidak pasti, format sama dengan *start*.
3. ***earliestEnd***, untuk akhir waktu yang tidak pasti, format sama dengan *start*.
4. ***end***, format sama dengan *start*.

5. **durationEvent**, *XML* dan *SPARQL*: "true" atau "false". *JSON*: "true" atau "false".
 - a) **true**, *event* terjadi dalam suatu durasi. Tidak memiliki *icon*. *Event* tergambar dengan pita biru. Warna pita ditetapkan dengan warna atribut. Warna *default*-nya adalah #58A0DC .
 - b) **false**, hanya fokus pada *event* pada tanggal tertentu (*instant event* yang memiliki *icon*). *Event* akan tergambar dalam icon biru (*default*) dengan pita biru pucat. Pita tersebut adalah warna *default* (atau warna atribut warna), dengan opasitas 20. Untuk merubah opasitas tersebut, ubah tema: {*impreciseOpacity*: 20} *value*. Maximum 100.
6. **title** , judul *text* yang beriringan dengan pita *timeline*. Juga dapat beriringan dengan *bubble* atau icon pada *timeline*. Atribut *title* ini *optional* (dapat dipakai ataupun ditinggalkan). Penjelasan akan muncul dalam sebuah *bubble* jika *event* ditekan.

2.2 **PHP (Hypertext Preprocessor)**

PHP (*Hypertext Preprocessor*) yang merupakan bahasa pemrograman berbasis *web* memiliki kemampuan untuk memproses data dinamis. *PHP* dikatakan sebagai sebuah *server-side embedded script language* artinya *sintaks-sintaks* dan perintah yang kita berikan akan sepenuhnya dijalankan oleh *server* tetapi disertakan pada halaman *HTML* biasa. Aplikasi-aplikasi yang dibangun oleh *PHP* pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *server*. (*PHP* dan *MySQL* Untuk Pemula, *MADCOMS*).

Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*. Dalam hal ini *client* menggunakan kode-kode *PHP* untuk mengirimkan permintaan ke *server*. Ketika menggunakan *PHP* sebagai *server-side embedded script language* maka *server* akan melakukan hal-hal sebagai berikut :

1. Membaca permintaan dari *client* / browser.
2. Mencari halaman / *page* di *server*.
3. Melakukan instruksi yang diberikan oleh *PHP* untuk melakukan modifikasi pada halaman / *page*.

4. Mengirim kembali halaman tersebut kepada *client* melalui internet atau intranet.

2.3 MySQL

Structured Query Language (SQL) adalah bahasa standar yang digunakan untuk mengakses *server database*. Semenjak tahun 70-an bahasa ini telah dikembangkan oleh *IBM*, yang kemudian diikuti dengan adanya *Oracle*, *Informix* dan *Sybase*. Dengan menggunakan *SQL*, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan misalnya *database* ataupun *Clipper* yang masih menggunakan perintah-perintah pemrograman murni.

MySQL adalah sebuah *server database SQL multiuser* dan *multi-threaded*. *SQL* sendiri adalah salah satu bahasa *database* yang paling populer di dunia. Implementasi program *server database* ini adalah program *daemon* '*MySQL*' dan beberapa program lain serta beberapa pustaka. *MySQL* dibuat oleh TcX dan telah dipercaya mengelola sistem dengan 40 buah *database* berisi 10,000 tabel dan 500 diantaranya memiliki 7 juta baris (kira-kira 100 *gigabyte* data). *Database* ini dibuat untuk keperluan sistem *database* yang cepat, handal dan mudah digunakan. Walaupun memiliki kemampuan yang cukup baik, *MySQL* untuk sistem operasi *Unix* bersifat *freeware*, dan terdapat versi *shareware* untuk sistem operasi *windows*.

Sistem *database MySQL* memiliki sistem sekuritas dengan tiga *verifikasi* yaitu *username*, *password* dan *host*. *Verifikasi host* memungkinkan untuk membuka sekuriti di '*localhost*', tetapi tertutup bagi *host* lain (bekerja di lokal komputer). Sistem sekuriti ini ada di dalam *database MySQL* dan pada tabel *user*. Proteksi juga dapat dilakukan terhadap *database*, tabel, hingga kolom secara terpisah.

2.4 Google Map

Google Map adalah layanan aplikasi peta *online* yang disediakan oleh *Google* secara gratis. Layanan peta *Google Maps* adalah layanan aplikasi peta *online* yang disediakan oleh *Google* secara gratis. Layanan peta *Google Maps*

secara resmi dapat diakses melalui situs <http://maps.Google.com>. Pada situs tersebut dapat dilihat informasi geografis pada hampir semua permukaan di bumi kecuali daerah kutub utara dan selatan. Layanan ini dibuat sangat interaktif, karena di dalamnya peta dapat digeser sesuai keinginan *user*, mengubah level *zoom*, serta mengubah tampilan jenis peta. *Google Maps* mempunyai banyak fasilitas yang dapat dipergunakan misalnya pencarian lokasi dengan memasukkan kata kunci, kata kunci yang dimaksud seperti nama tempat, kota, atau jalan, fasilitas lainnya yaitu perhitungan rute perjalanan dari satu tempat ke tempat lainnya.

2.4.1 Cara kerja Google Map

Google Maps dibuat dengan menggunakan kombinasi dari gambar peta, *database*, serta obyek-obyek interaktif yang dibuat dengan bahasa pemrograman *Hypertext Markup Language (HTML)*, *Javascript* dan *AJAX*, serta beberapa bahasa pemrograman lainnya. Gambar-gambar yang muncul pada peta merupakan hasil komunikasi dengan *database* pada *web server Google* untuk menampilkan gabungan dari potongan-potongan gambar yang diminta. Keseluruhan citra yang ada diintegrasikan ke dalam *database* pada *Google Server*, yang nantinya akan dapat dipanggil sesuai kebutuhan permintaan. Bagian-bagian gambar map merupakan gabungan dari potongan gambar-gambar bertipe PNG yang disebut *tile* yang berukuran 256 x 256 pixel seperti gambar 2.4



Gambar 2.4 Potongan gambar peta di *Google Map*

Tiap-tiap potongan gambar diatas, mewakili gambar tertentu dalam *longitude*, *latitude* dan *zoom* level tertentu. Kode *Javascript* yang digunakan untuk menampilkan

peta *Google Maps* diambil dari link *URL*. Jadi untuk menampilkan peta suatu lokasi yang diinginkan, dapat dengan cara mengirimkan *URL* yang diinginkan, misalnya:
`http://maps.Google.com/?ie=UTF8&ll=6.500899,106.918945&spn=4.327078,4.938354&z=8`

1. `ie=UTF8`, adalah karakter *encoding* untuk *map*.
2. `ll=-6.500899,106.918945`, adalah posisi titik tengah peta yaitu *latitude* (lintang) dan *longitude* (bujur) dari peta yang ditampilkan, pada *link* diatas posisi titik tengah peta pada *latitude*: -6.500899 dan *logitude*: 106.918945.
3. `spn=4.327078,4.938354`, adalah rentang dari *latitude* dan *longitude*-nya.
4. `z=8`, adalah tingkatan/level *zoom* peta.

2.4.2 Google Map API

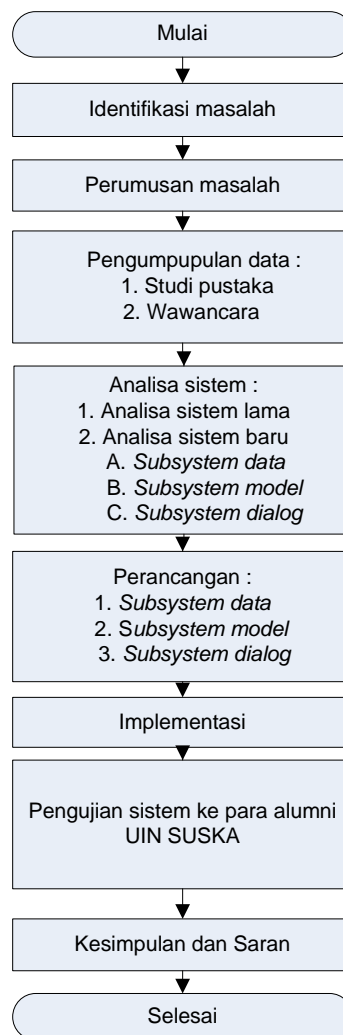
API atau *Application Programming Interface* merupakan suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Dengan adanya *API* ini, maka memudahkan *programmer* untuk “membongkar” suatu *software* untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. *API* dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan *programmer* menggunakan sistem *function*. Proses ini dikelola melalui *operating system*.

Keunggulan dari *API* ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi. Bahasa pemrograman yang digunakan oleh *Google Maps* yang terdiri dari *Hypertext Markup Language (HTML)*, *Javascript* dan *AJAX* serta *XML*, memungkinkan untuk menampilkan peta *Google Maps* di *website* lain.

BAB III

METODOLOGI PENELITIAN

Pada bab ini akan dipaparkan tentang langkah-langkah yang digunakan untuk membahas permasalahan yang diambil dalam penelitian atau yang disebut dengan metodologi penelitian. Metodologi penelitian tugas akhir ini dapat digambarkan seperti yang terlihat pada gambar 3.1 sebagai berikut:



Gambar 3.1 *Flowchart* Metodologi Penelitian

3.1 Identifikasi Masalah

Identifikasi masalah merupakan sekelompok aspek yang berada disekitar masalah utama yang dapat diteliti untuk menjawab permasalahan utama. Adapun permasalahan yang dapat diidentifikasi untuk pelaksanaan tugas akhir ini adalah perhitungan yang masih manual.

3.2 Perumusan Masalah

Berdasarkan permasalahan yang telah diidentifikasi, maka dapat dirumuskan bahwa bagaimana merancang dan membangun suatu sistem Visualisasi Sebaran Alumni Universitas Islam Negri Sultan Syarif Kasim Riau.

3.3 Pengumpulan Data

3.3.1 Studi Pustaka

Studi pustaka dilakukan untuk mencari dan mempelajari serta mengumpulkan seluruh informasi yang terkait dan mendukung pelaksanaan penelitian pada tugas akhir ini. Studi pustaka ini membahas tentang Visualisasi Sebaran Alumni Universitas Islam Negri Sultan Syarif Kasim Riau. Sumber kepustakaan diambil dari karya ilmiah yang berasal dari buku-buku maupun internet. Karya ilmiah yang dimaksud adalah berupa tulisan ilmiah yang berbentuk artikel, prosiding, buku, *e-book* (buku elektronik), dan lain-lain.

3.4 Analisa Sistem

3.4.1 Analisa Penelitian sebelumnya

The “American Century” Geospasial Timeline, sebuah aplikasi *web* yang dibangun oleh Brian Croxall, seorang *Visiting Assistant Profesor* Departemen Bahasa Inggris di *Clemson University, South Carolina*, Amerika Serikat. Aplikasi ini diadaptasi dari *“Timeline Assignment”* Jason B. Jones, seorang *Associate Professor of English* di *Central Connecticut State University, Connecticut*, Amerika Serikat. Data dikumpulkan oleh mahasiswa Croxall di kelas *English 399* pada musim semi tahun 2010, menggunakan proyek milik *SIMILE* yaitu exhibit dan timeline.

Croxall (2010) menyatakan bahwa beberapa tahun terakhir dia memiliki ketertarikan dalam merepresentasikan waktu dan lokasi secara visual dalam konteks program sastra. Dalam tahun akhirnya sebagai mahasiswa pascasarjana, Croxall merupakan anggota *Emory's Center for Interactive Teaching*, disana Croxall menjelajahi widget-widjet web yaitu widget-widjet hasil dari *SIMILE project* yang dikembangkan oleh MIT.

Untuk melakukan penelitian lebih lanjut, Croxall menghubungi Jason B. Jones yang diketahui Croxall telah dahulu kenal dengan *SIMILE project* melalui blog yang dimilikinya. Jones saat itu sedang mengajar tentang Era *Victoria*, Jones langsung membuat tugas *timeline* yang diprogram oleh Croxall, pekerjaan gabungan ini telah ditulis Jones dalam blognya.

Pada saat mengajar pada kelas survey Croxall melakukan representasi data melalui *timeline* dengan data hasil survey yang dilakukan mahasiswanya tentang perkembangan Sastra di Amerika setelah masa perang sipil, tugas ini diangkat seutuhnya dari tugas yang dimiliki Jones. Croxall tidak hanya ingin merepresentasikan data visual berbasis waktu saja, namun dia juga ingin merepresentasikan data visual berbasis lokasi, maka Croxall mempelajari integrasi antara *Google Map* dengan *SIMILE timeline* dengan tugas survey yang diberikan kepada mahasiswanya, Croxall telah berhasil membuat aplikasi yang memberikan penambahan dari tugas yang dimiliki Jones dengan memasukkan *element geospasial*.



Gambar 3.2 Tampilan Aplikasi *American Century*.

Sumber: <http://www.brianCroxall.net/timelines/AmericanSurveyTimeline2010.html>

Aplikasi “*American Century*” memiliki data *filter* berupa *event type* berupa kategori seperti seni, ekonomi, literatur, dan lain sebagainya. *Filter* data juga bisa diberikan dalam dekade tertentu, dapat dilihat seperti gambar 3.3:



Gambar 3.3 Tampilan menu data filter.

Hasil data *filter* dapat tampil dalam peta *geospasial*, seperti terlihat pada gambar 3.4:



Gambar 3.4 Tampilan data dalam peta *geospasial*. Sumber:

<http://www.brianCroxall.net/timelines/AmericanSurveyTimeline2010.html>

3.4.2 Analisa Sistem Baru

Analisa sistem baru dilakukan untuk menyusun langkah-langkah dalam mengidentifikasi permasalahan-permasalahan yang akan terjadi pada sistem yang akan dibangun. Serta kebutuhan-kebutuhan apa saja yang diinginkan untuk mengatasi permasalahan yang ada pada sistem nantinya. Dalam analisa sistem terdiri atas beberapa subsistem, diantaranya adalah sebagai berikut :

3.4.2.1 Subsistem Data

Analisa subsistem data merupakan sebuah gambaran *database* yang akan dibuat pada aplikasi terdiri atas masukan data dan keluaran data. Analisa ini digambarkan dalam bentuk *Entitas Relational Diagram (ERD)*, yang pada kelanjutannya akan mengacu dalam perancangan *database* secara keseluruhan.

3.4.2.2 Subsistem Model

Dalam perancangan aplikasi yang akan dibangun, aplikasi hanya dapat menghitung nilai dari pembobotan dan perbandingan yang dilakukan oleh seorang pembuat keputusan, pengisian tersebut meliputi kriteria yang mendukung pemilihan *vendor*. Hasil yang akan di dapat berupa hasil dari Visualisasi Sebaran Alumni Universitas Islam Negri Sultan Syarif Kasim Riau.

3.4.2.3 Subsistem Dialog

Analisa pada subsistem dialog digambarkan dengan *Data Flow Diagram (DFD)*, yang pada akhirnya akan mengacu dalam perancangan struktur menu dan *User Interface*.

3.5 Perancangan

3.5.1 Subsistem Data

Tahap perancangan subsistem data merupakan hasil dari analisa data yaitu *ERD*, yang selanjutnya akan dibuat suatu perancangan tabel secara utuh dan lengkap dengan berbagai komponennya.

3.5.2 Subsistem Model

Perancangan model merupakan hasil dari analisa model yaitu metode yang digunakan dalam pembuatan aplikasi tersebut. Pada subsistem ini akan dibuat suatu desain model *system* berupa *Flowchart* dari proses Visualisasi Sebaran Alumni Universitas Islam Negri Sultan Syarif Kasim Riau

3.5.3 Subsistem Dialog

Perancangan subsistem dialog akan menghasilkan sebuah perancangan struktur menu aplikasi dan desain *User Interface* pada aplikasi, yang diperoleh dari analisa subsistem dialog atau implementasi dari analisa *DFD*.

3.6 Implementasi

Implementasi sistem merupakan hasil dari desain sistem yang telah dirancang kemudian diimplementasikan pada sebuah program komputer. Bahasa pemrograman yang digunakan adalah *PHP*.

3.7 Pengujian

Tahap pengujian diperlukan untuk menjadi ukuran bahwa sistem dapat dijalankan sesuai dengan tujuan, yang akan dilakukan dengan cara sebagai berikut:

Setelah sistem ini selesai dibuat, akan dilakukan pengujian dengan menggunakan data-data alumni yang sudah tersimpan pada jurusan tertentu, dan menjalankan fungsi aplikasi yang telah dibangun sesuai perancangan, untuk mengetahui apakah sistem sudah mampu memberikan visualisasi data alumni yang ada.

3.8 Kesimpulan dan Saran

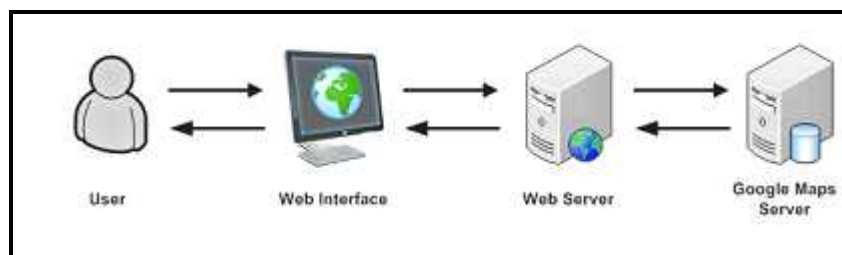
Kesimpulan ini merupakan kesimpulan dari suatu pembahasan untuk menjawab pertanyaan-pertanyaan yang dikemukakan pada masalah dan tujuan serta saran-saran yang dikemukakan.

BAB IV

ANALISA DAN PERANCANGAN

Perancangan dalam pembuatan Tugas Akhir ini meliputi *Desain Arsitektur Sistem*, *Data flow diagram*, *Entity Relationship Diagram (ERD)*, *Flowchart*, dan *Site flow Diagram*.

4.1 *Desain Arsitektur Sistem*



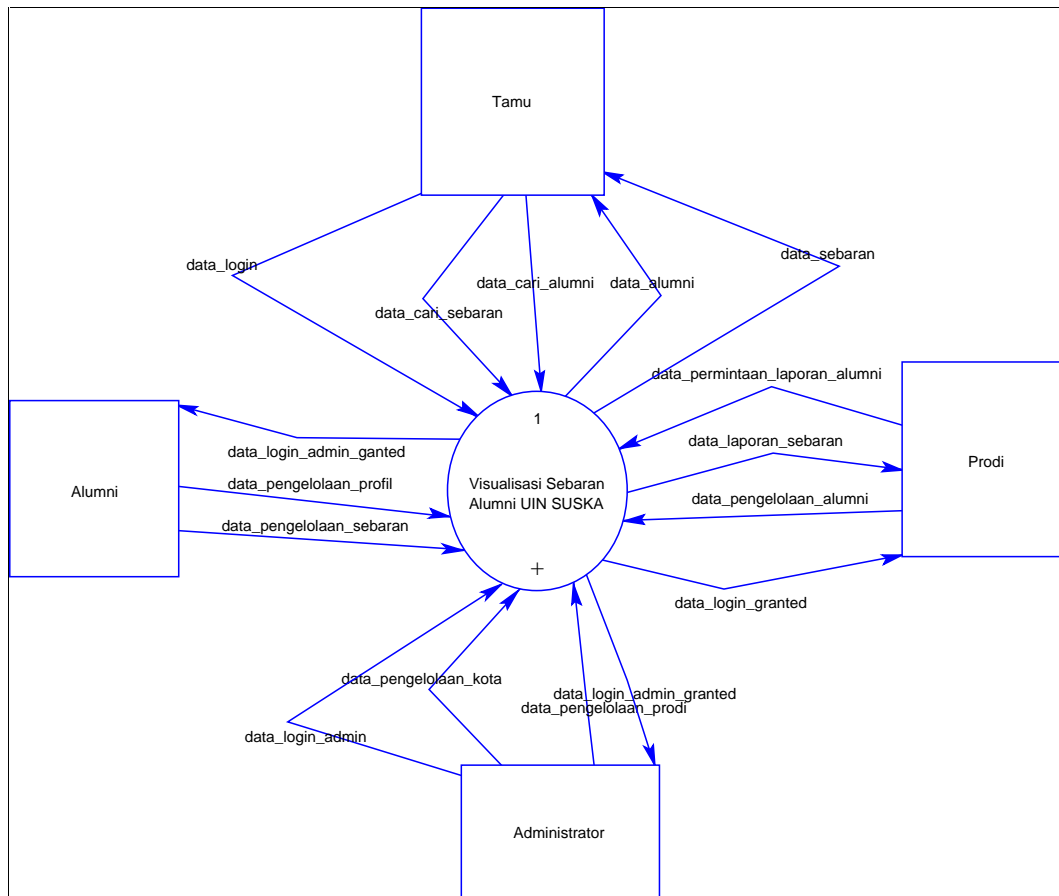
Gambar 4.1 Arsitektur Sistem Visualisasi Sebaran Alumni UIN SUSKA

Berdasarkan gambar 4.1 diatas *user* berkomunikasi dengan sistem melalui *web browser*, apabila situs web ini dibuka, maka *browser* akan menampilkan konten web dari situs yang terdapat pada *web server*. Aplikasi web inilah yang akan berinteraksi secara interaktif dengan *user*, apabila *user* melakukan suatu perintah, maka eksekusinya akan diproses di *browser* atau *web server*, dan apabila terdapat permintaan dari aplikasi untuk mengakses *database*, maka *database* tersebut akan dipanggil ke dalam program yang diambil dari *web server*, lalu dilakukan *request* data yang diminta ke *server* Google Maps. Hasilnya adalah berupa gambar peta, serta objek-objek yang dimiliki oleh peta Google Maps yang selanjutnya akan dikembalikan ke *web browser* berupa tampilan peta yang memiliki point- point lokasi yang diminta didalamnya.

4.2 *Context Diagram*

Context Diagram menggambarkan bagaimana *external entity* berinteraksi terhadap sistem melalui aliran data, disebut juga *top level DFD*, yaitu diagram yang paling tidak *detail*, dari sebuah sistem informasi yang menggambarkan

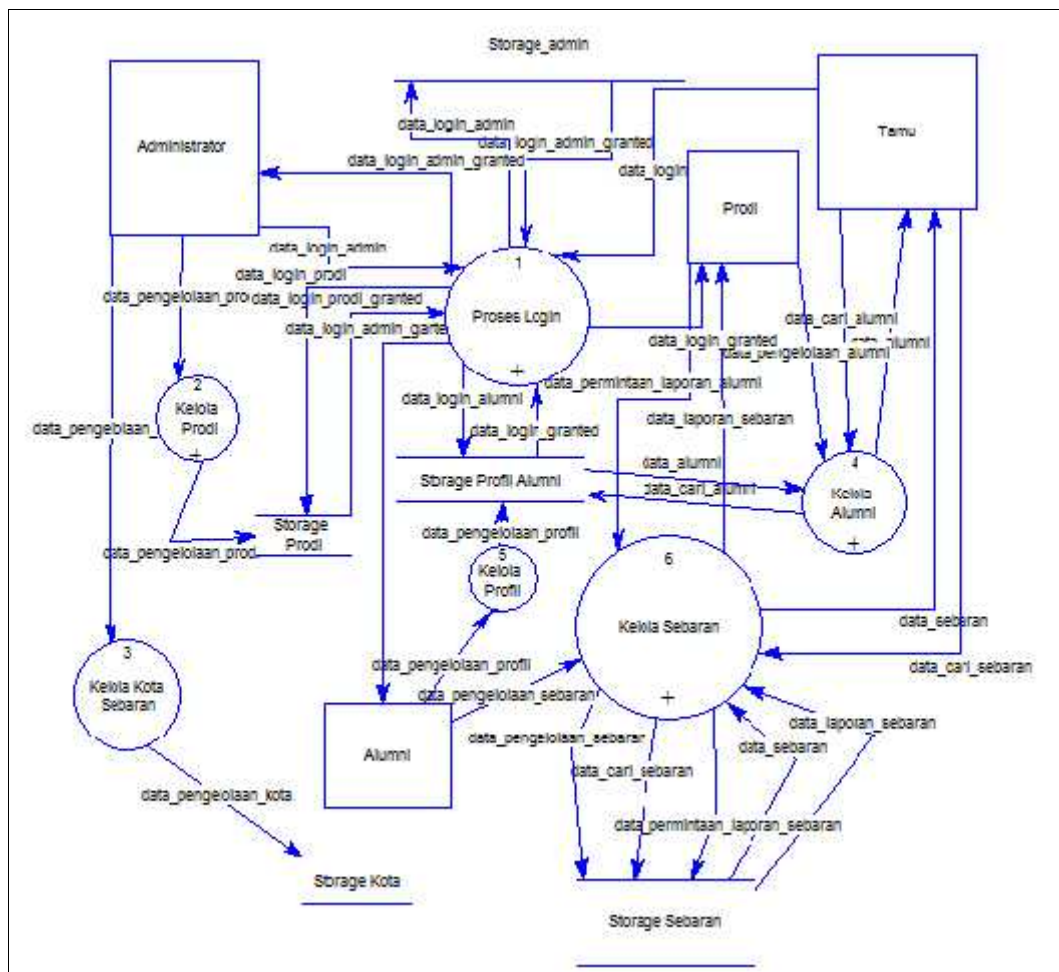
aliran-aliran data ke dalam dan ke luar sistem dan ke dalam dan ke luar *external entity*. Seperti terlihat pada gambar 4.2.



Gambar 4.2 Context Diagram Sistem

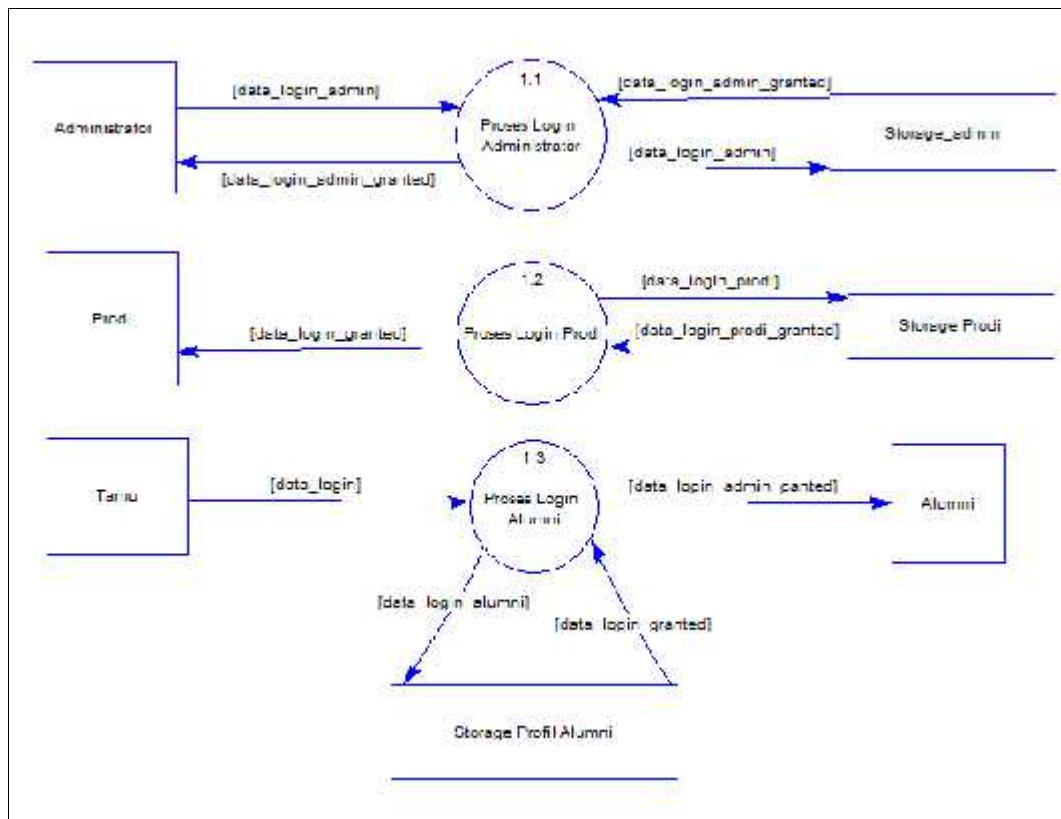
4.3 Data Flow Diagram Level 1

Data flow diagram merupakan diagram aliran data yang berjalan dalam sistem dari *entity* ke proses, proses ke penyimpanan data, hingga proses yang mengembalikan aliran data ke *entity*, DFD menggambarkan hubungan antar subsistem. Seperti terlihat pada gambar 4.3.



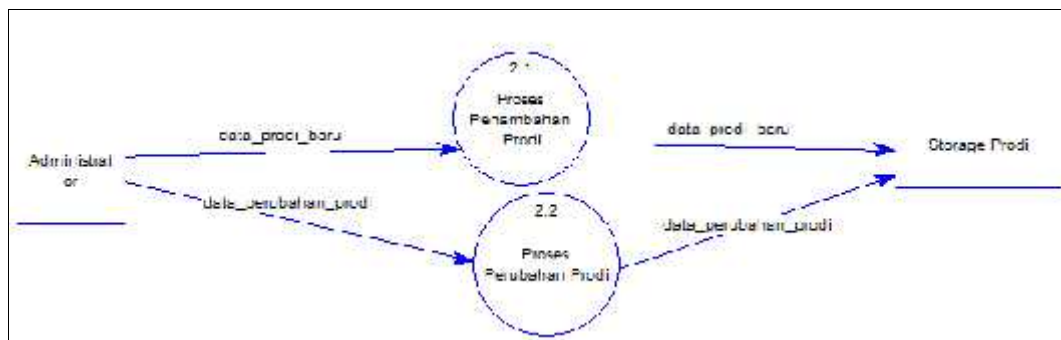
Proses-proses yang lebih rinci dapat dilihat pada DFD hasil *breakdown* proses-proses yang ada pada DFD level 1.

4.4.1 DFD Level 2.1



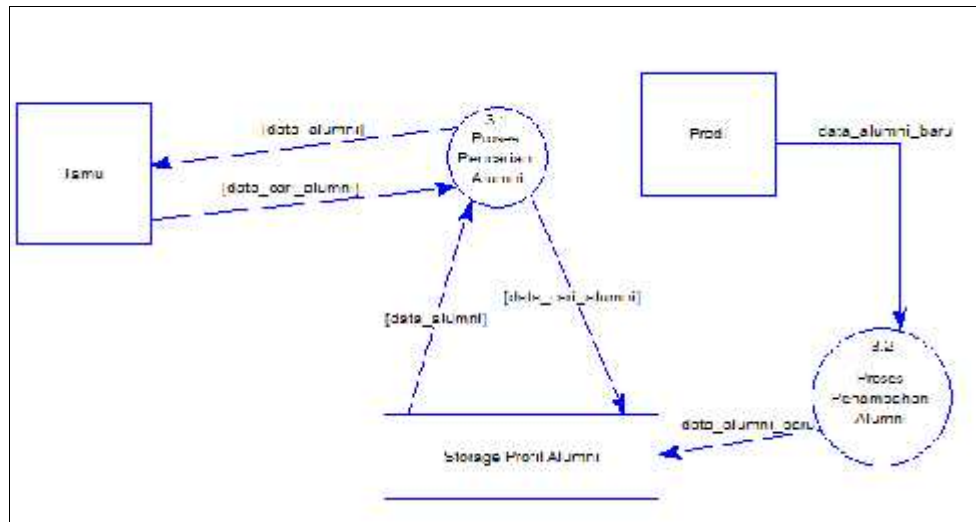
Gambar 4.4 DFD Level 2.1 : Proses *Login*

4.4.2 DFD Level 2.2



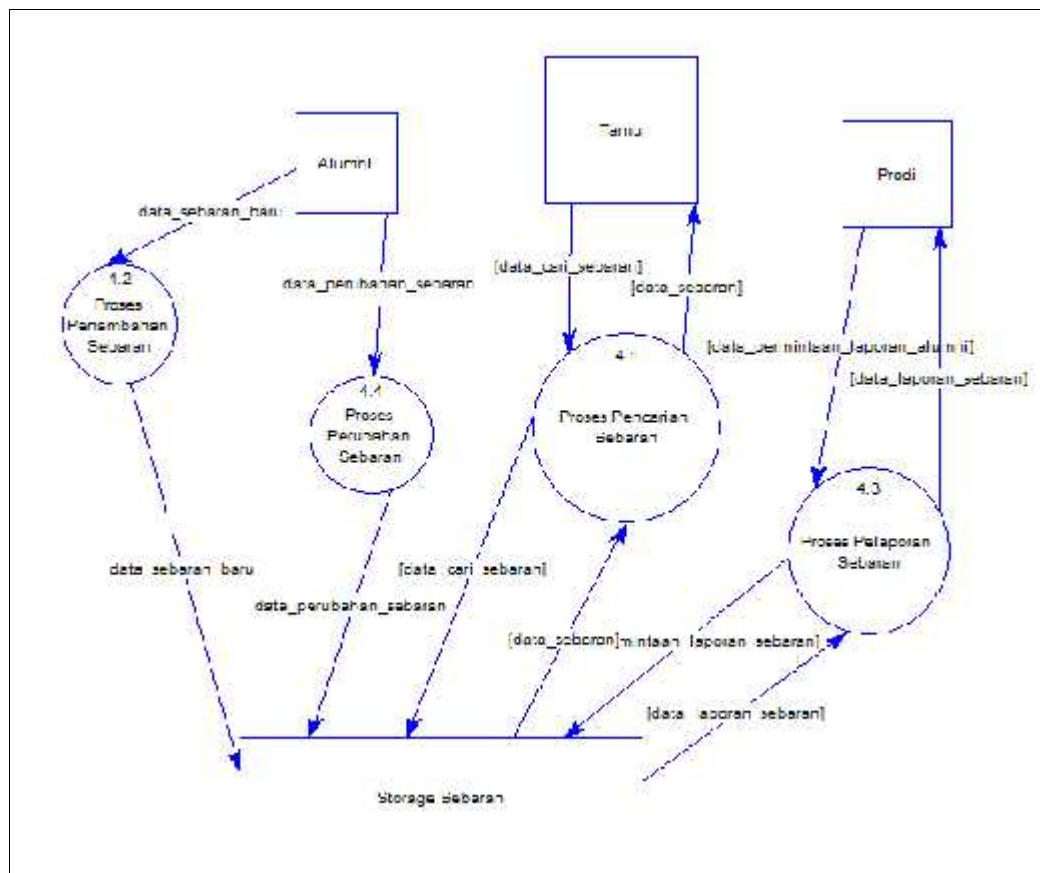
Gambar 4.5 DFD Level 2.2 : Pengelolaan Prodi

4.4.3 DFD Level 2.3



Gambar 4.6 DFD Level 2.3 : Pengelolaan Alumni

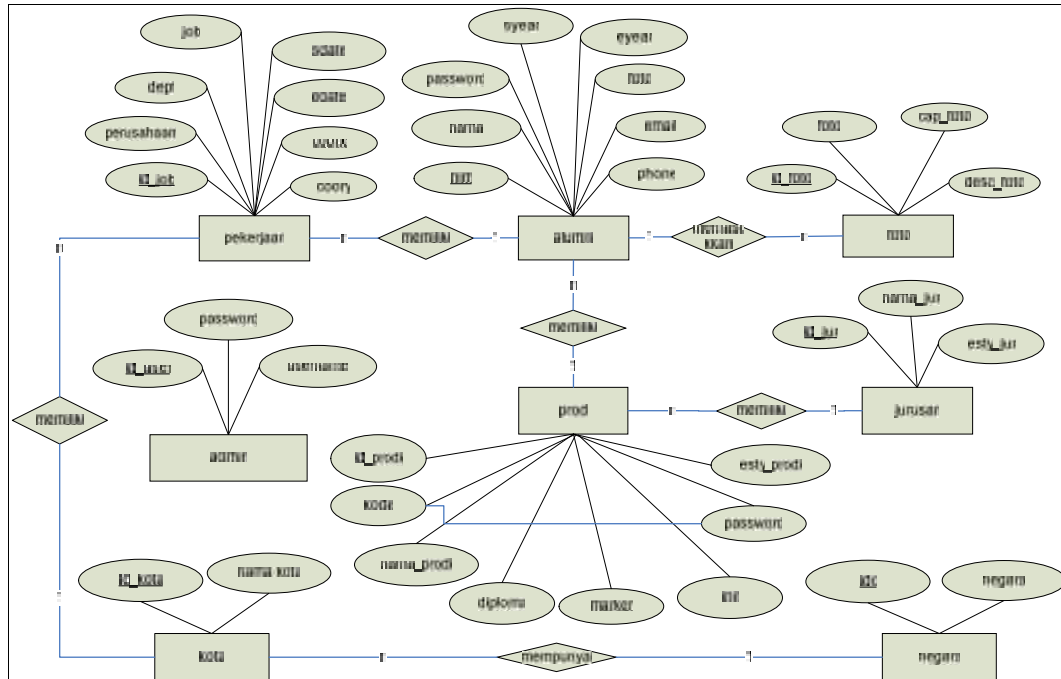
4.4.4 DFD Level 2.4



Gambar 4.7 DFD Level 2.4 : Pengelolaan Sebaran.

4.5 Entity Relationship Diagram

ERD menggambarkan hubungan antara *entity* yang mewakili tabel-tabel pada *database* beserta atribut-atributnya, *entity* yang berhubungan adalah jurusan, prodi, alumni, pekerjaan, dan foto, sementara tabel negara berelasi dengan tabel kota.



Gambar 4.8 ERD sistem.

Tabel – tabel yang terbentuk berdasarkan *ERD* sistem adalah :

1. Tabel *admin*

Tabel ini menyimpan data-data *Administrator*.

Tabel 4.1 Tabel *admin*

Atribut	Tipe Data	keterangan
<i>id_user</i>	Varchar(6)	<i>Primary key</i> , Menyimpan id user admin.
<i>password</i>	Varchar(8)	Menyimpan password user admin.
<i>username</i>	Varchar(12)	Menyimpan nama user admin.

2. Tabel negara

Tabel ini menyimpan data-data kota sebaran.

Tabel 4.2 Tabel negara

Atribut	Tipe Data	keterangan
idc	Int (3)	<i>Primary key</i> , Menyimpan id negara
negara	Varchar(225)	Menyimpan nama negara

3. Tabel kota

Tabel ini menyimpan data-data kota sebaran.

Tabel 4.3 Tabel kota

Atribut	Tipe Data	keterangan
Id_kota	Int(3)	<i>Primary key</i> , Menyimpan id kota
nama_kota	Varchar(225)	Menyimpan nama kota
idc	Varchar(3)	<i>Foreign Key</i> , Menyimpan id negara

4. Tabel jurusan

Tabel ini menyimpan data-data Jurusan.

Tabel 4.4 Tabel jurusan.

Atribut	Tipe Data	keterangan
id_jur	Varchar(6)	<i>Primary key</i> , Menyimpan id jurusan.
nama_jur	Varchar(50)	Menyimpan <i>password user</i> .
esty_jur	Date	Menyimpan tahun berdiri jurusan.

5. Tabel prodi

Tabel ini menyimpan data-data Prodi.

Tabel 4.5 Tabel prodi

Atribut	Tipe Data	keterangan
id_prodi	Varchar(6)	<i>Primary key</i> , Menyimpan id prodi
kode	Varchar(8)	Menyimpan kode prodi
nama_prodi	Varchar(50)	Menyimpan nama prodi
diploma	Varchar(1)	Menyimpan jenis program studi
init	Varchar(3)	Menyimpan insial prodi
<i>marker</i>	Varchar(255)	Menyimpan nama <i>marker</i> prodi
<i>password</i>	Varchar(8)	Menyimpan <i>password</i> prodi.
esty_prodi	Date	Menyimpan tahun berdiri prodi.
id_jur	Varchar(6)	<i>Foreign Key</i> , Menyimpan id jurusan.

6. Tabel alumni

Tabel ini menyimpan data-data pribadi alumni.

Tabel 4.6 Tabel alumni

Atribut	Tipe Data	keterangan
nim	Varchar(16)	<i>Primary key</i> , Menyimpan Nomor Induk Mahasiswa alumni.
nama	Varchar(50)	Menyimpan nama alumni
sex	Varchar(15)	Menyimpan jenis kelamin alumni
<i>password</i>	Varchar(8)	Menyimpan <i>password</i> alumni
syear	Date	Menyimpan tahun masuk alumni.
eyear	Date	Menyimpan tahun lulus alumni
email	Varchar (50)	Menyimpan alamat Email alumni
phone	Varchar (15)	Menyimpan nomor telepon alumni
foto	Varchar(255)	Menyimpan nama foto alumni
id_prodi	Varchar(6)	<i>Foreign Key</i> , menyimpan id prodi

7. Tabel pekerjaan

Tabel ini menyimpan data-data sebaran alumni dalam waktu dan tempat tertentu.

Tabel 4.7 Tabel pekerjaan.

Atribut	Tipe Data	Keterangan
id_job	Varchar(15)	<i>Primary key</i> , Menyimpan ID perusahaan tempat kerja alumni.
perusahaan	Varchar(50)	Menyimpan nama perusahaan tempat kerja alumni.
dept	Varchar(50)	Menyimpan nama departemen perusahaan tempat kerja alumni.
job	Varchar(30)	Menyimpan jabatan alumni.
idk	Int (3)	<i>Foreign Key</i> , menyimpan ID Kota.
sdate	Date	Menyimpan tanggal masuk alumni bekerja.
edate	Date	Menyimpan tanggal selesai alumni bekerja
coorx	Varchar (15)	Menyimpan nilai koordinat x
coory	Varchar(15)	Menyimpan nilai koordinat y
nim	Varchar(15)	<i>Foreign Key</i> , NIM milik alumni.

8. Tabel Foto

Tabel ini menyimpan data-data foto alumni yang berhubungan dengan pekerjaannya.

Tabel 4.8 Tabel foto

Atribut	Tipe Data	Keterangan
id_foto	Varchar (15)	<i>Primary key</i> , Menyimpan ID foto.
foto	Blob	Menyimpan file foto
cap_foto	Varchar (50)	Menyimpan caption foto
desc_foto	Varchar varchar (225)	Menyimpan deskripsi foto
nim	Varchar (15)	<i>Foreign Key</i> , Menyimpan NIM milik alumni.

4.6 Flowchart Sistem

Flowchart merupakan suatu metode yang digunakan untuk menggambarkan alur suatu program secara lebih mudah dan sederhana, *flowchart* menunjukkan rangkaian kegiatan-kegiatan program dari awal hingga akhir. Berikut adalah beberapa *flowchart* yang terdapat pada sistem. Seperti terlihat pada gambar 4.9.

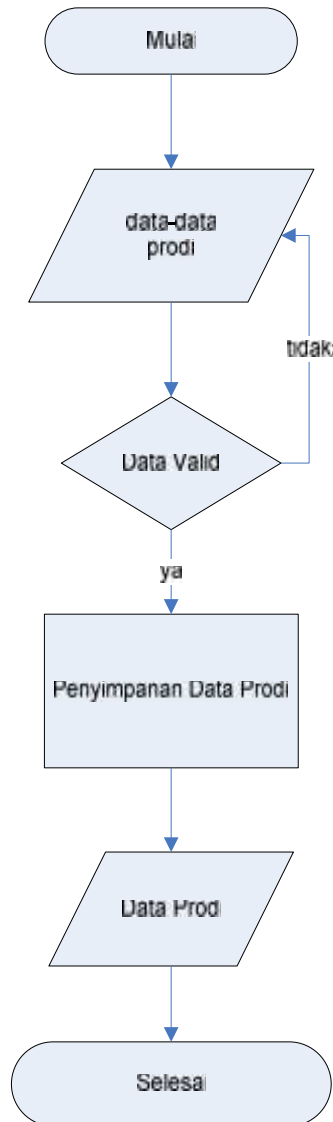
1. *Flowchart* pembuatan sistem.



Gambar 4.9 *Flowchart* penambahan prodi oleh admin.

Flowchart ini menggambarkan alur pembuatan sistem dari tahap awal hingga sistem ini dapat dijalankan untuk melakukan pengujian kepada user.

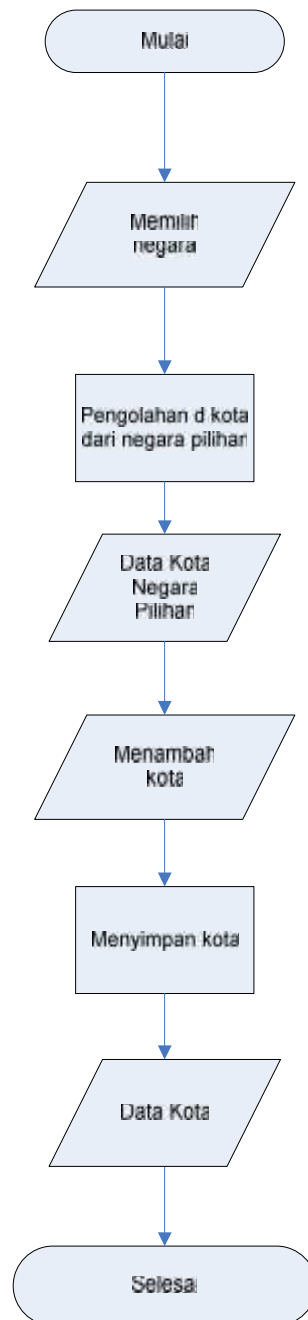
2. *Flowchart* penambahan prodi oleh *admin*.



Gambar 4.10 *Flowchart* penambahan prodi oleh *admin*.

Berdasarkan gambar 4.10 diatas *administrator* adalah pihak yang mengelola data prodi, *administrator* yang menambahkan data prodi pada sistem, penambahan berarti mendaftar sebuah *user* prodi untuk dapat mengakses sistem (*login*), data prodi yang dimasukkan adalah, id Prodi, nama, kode, jenis diploma, tanggal berdiri, inisial, dan *marker*. Prodi *login* untuk menjalankan perannya sebagai pihak yang mengelola data alumni.

3. *Flowchart* penambahan kota oleh *admin*.

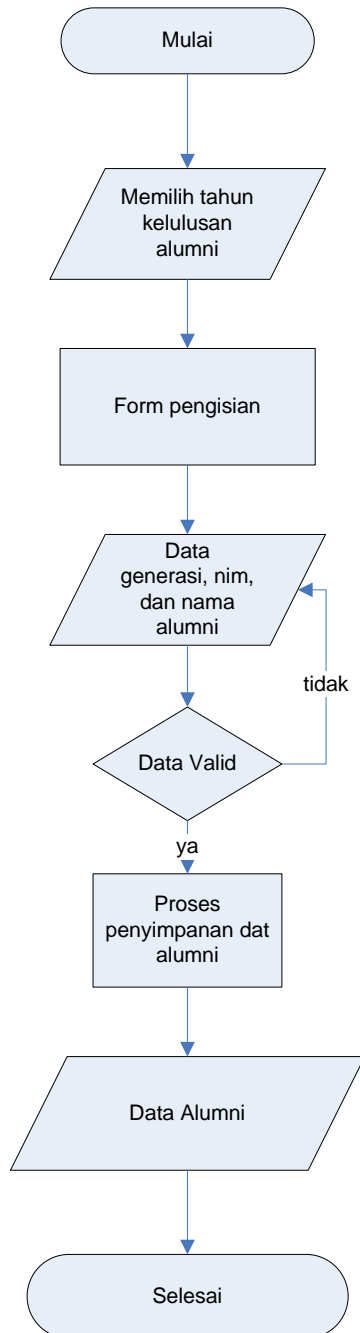


Gambar 4.11 *Flowchart* penambahan kota oleh *admin*.

Berdasarkan gambar 4.11 diatas *administrator* juga berperan sebagai pihak yang menyediakan data-data kota yang nantinya dipakai oleh alumni sebagai data sebaran, fungsi pengelolaan kota sebaran ini adalah untuk menencegah error, sebagai pengganti pengetikan data kota secara manual oleh alumni, dengan

menyediakan data kota, maka ketidakkonsistensian pengisian data kota bisa dicegah.

4. *Flowchart* penambahan data alumni oleh prodi.

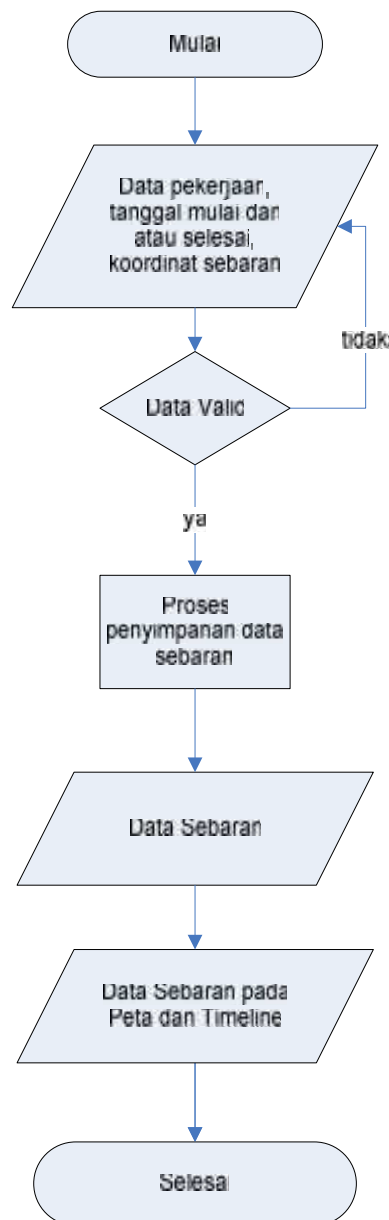


Gambar 4.12 *Flowchart* penambahan data alumni oleh prodi.

Berdasar gambar 4.12 diatas salah satu peran *user* prodi adalah menambahkan data alumninya masing-masing dengan menentukan tahun

kelulusannya terlebih dahulu, *user* prodi mendaftarkan alumni ke sistem, agar alumni dapat mengakses, mengisi dan memperbarui data sebaran mereka.

5. *Flowchart* penambahan data sebaran oleh alumni.



Gambar 4.13 *Flowchart* penambahan data alumni oleh prodi.

Berdasarkan gambar 4.13 diatas, setelah didaftarkan oleh *admin*, alumni dapat mengakses profilnya dengan *login* menggunakan *nim* dan *password* yang diberikan, alumni mengisi data pekerjaan, koodinat lokasi, tanggal mulai pekerjaan atau tanggal mulai dan akhir pekerjaan, bila pekerjaan sudah selesai,

data pekerjaan ini disimpan dan digunakan sebagai data sebaran yang nantinya ditampilkan dalam peta geospasial dan *timeline*.

6. *Flowchart* pembuatan visualisasi peta geospasial

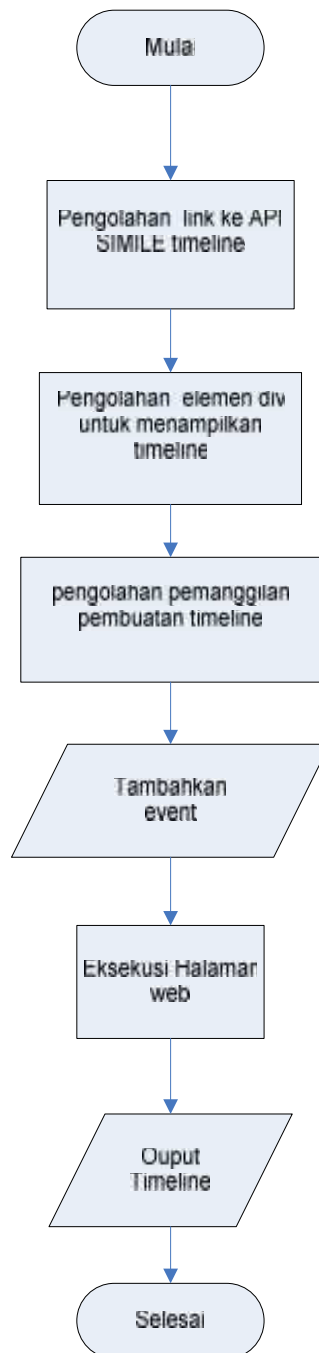


Gambar 4.14 *Flowchart* pembuatan visualisasi peta geospasial

Berdasarkan gambar 4.14 diatas, untuk visualisasi peta geospasial Google map, terlebih dahulu masukkan google API dengan menautkan pada sebuah

JavaScript, lalu elemen div dibuat sebagai tempat peletakkan peta pada web, selanjutnya obyek-obyek peta ditulis, lalu buat fungsi JavaScript untuk membuat peta, dan terkakhir inisialisasi pada *body* halaman web, untuk menampilkan visualisasi, eksekusi halaman HTML pada web *browser*.

7. *Flowchart* pembuatan visualisasi *timeline*



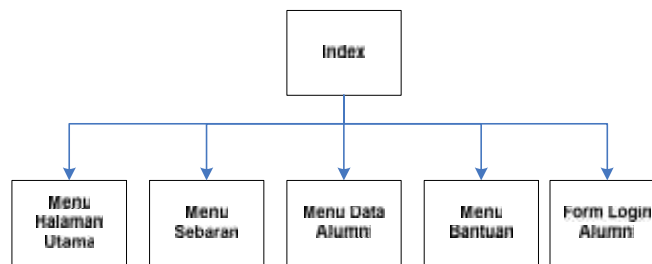
Gambar 4.15 *Flowchart* pembuatan visualisasi *timeline*

Berdasarkan gambar 4.15 diatas, untuk visualisasi *timeline*, terlebih dahulu lakukan pentautan ke API SIMILE *timeline*, buat elemen div pada halaman HTML untuk menempatkan *timeline*, buat pemanggilan untuk melakukan pembuatan *timeline*, tambahkan *event* sebagai data waktu yang akan ditampilkan, dan proses eksekusi di web *browser* akan menampilkan *event timeline* pada halaman web.

4.7 Site flow Diagram

Site flow Diagram menggambarkan alur menu yang dapat ditelusuri dalam sistem.

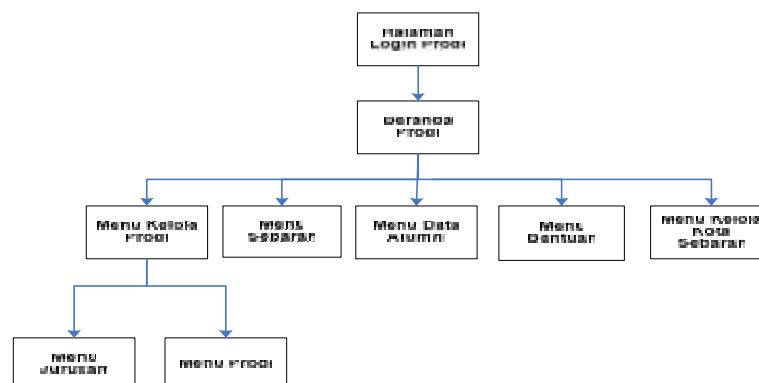
1. Site flow Tamu



Gambar 4.16 Site flow Tamu

Tamu atau *Public User* dapat menjelajahi web, fitur yang diberikan kepada Public User adalah, menu sebaran yang menampilkan sebaran alumni pada peta beserta *timeline*, menu data alumni yang melihatkan data-data alumni pada sistem, menu bantuan dan tombol yang bertautan dengan halaman *login* alumni untuk alumni yang ingin mengakses profil dan data sebarannya.

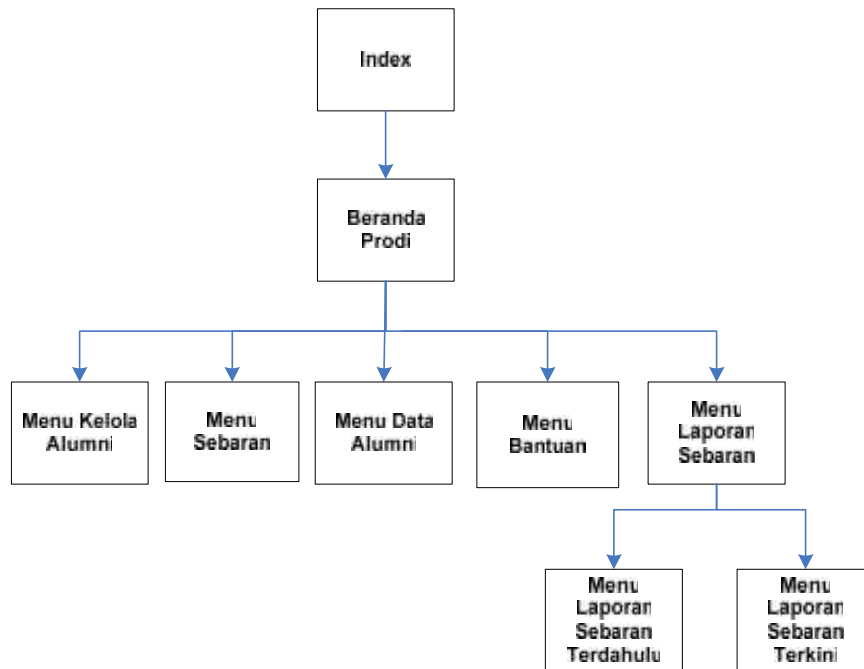
2. Site flow Administrator.



Gambar 4.17 Site flow Administrator

Berdasarkan gambar 4.17 diatas, *administrator* bertugas untuk mengelola prodi dan kota sebaran, sehingga fitur web yang disediakan untuk *administrator* meliputi menu kelola prodi, dan menu kelola kota sebaran, fitur umum seperti sebaran, data alumni, dan bantuan masih disertakan untuk *administrator*.

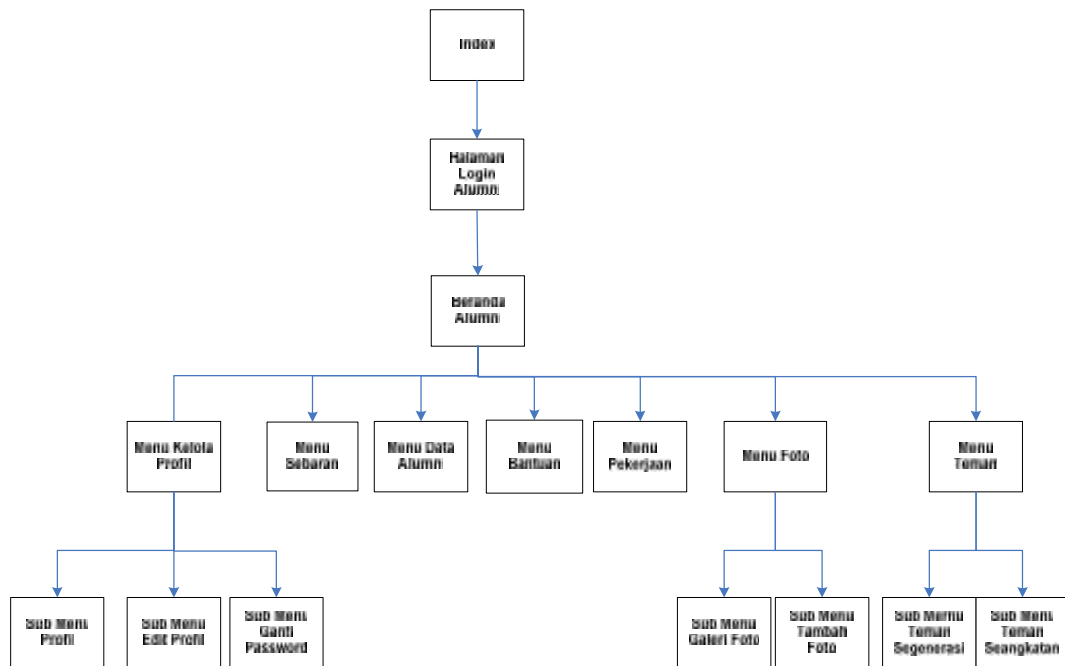
3. *Site flow* Prodi



Gambar 4.18 *Site flow* Prodi

Berdasarkan gambar 4.18 diatas, prodi bertugas untuk mengelola alumni dan memantau sebaran, sehingga fitur web yang disediakan untuk prodi meliputi menu kelola alumni dan menu laporan sebaran alumni untuk sebaran terawal (sebaran pertama setelah lulus dari UIN SUSKA) dan sebaran terkini, fitur umum seperti sebaran, data alumni, dan bantuan masih disertakan untuk prodi.

4. *Site flow* Alumni



Gambar 4.19 *Site flow* Alumni

Berdasarkan gambar 4.19 diatas, alumni dapat masuk ke sistem jika telah didaftarkan oleh prodi, alumni dapat mengelola profilnya sendiri dengan mengisi data-data yang masih kosong atau menambahkan foto profil sehingga *user* lain baik publik ataupun terdaftar dapat melihat identitas seorang alumni, untuk dapat dilihat dalam sebaran alumni UIN SUSKA di peta geospasial *Google Map* dan *Timeline SIMILE*, alumni harus mengisi pekerjaan terkini dengan memasukkan tanggal mulai tanpa tanggal selesai, setelah data masuk akan ada *marker* yang mewakili keberadaan alumni, selain itu alumni dapat menamahkan foto bila perlu, untuk memparkaya informasi sebaran yang dimiliki oleh alumni.

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi

Sistem yang dibangun diharapkan dapat melakukan visualisasi sebaran alumni UIN Suska Riau dengan menampilkan lokasi seorang alumni secara langsung dalam sebuah peta dan melihatkan titik dan interval waktu dalam sebuah timeline. Visualisasi Sebaran Alumni UIN SUSKA RIAU menggunakan bahasa pemrograman *PHP*, dan database *MySQL* untuk menyimpan semua data, sementara dalam melakukan visualisasi sebaran, sistem menggunakan layanan yang disediakan *Google Maps* dan *Simile timeline* yang diintegrasikan dengan *PHP* dan *MySQL* untuk mengolah data yang tersimpan.

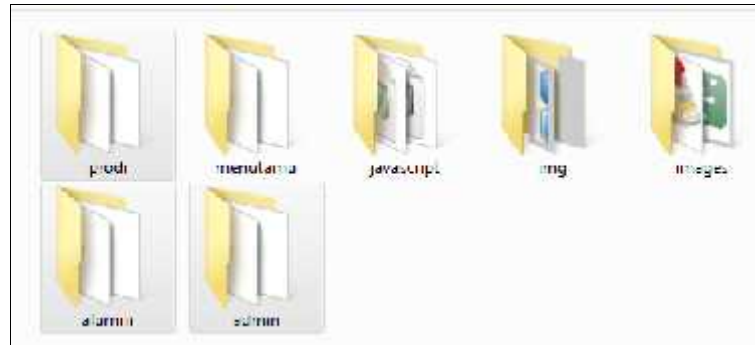
5.1.1 Kebutuhan Sistem

Sistem yang dibangun memiliki kebutuhan yang diperlihatkan dalam Tabel 5.1

Tabel 5.1 Kebutuhan Sistem

Kebutuhan	Spesifikasi
Komputer	Sistem Operasi Windows 7
Layanan <i>Google Maps</i> Api	Diambil dari : http://maps.google.com/maps/api/js
Layanan <i>Simile Timeline</i>	Diambil dari : http://api.simile-widgets.org/timeline/2.3.1/timeline-api.js
Tools	<ol style="list-style-type: none">1. Compiler Macromedia Dreamweaver untuk menulis kode-kode HTML dan <i>PHP</i>.2. <i>Web Server</i> untuk <i>Windows OS</i> Wamp server yang sudah memiliki Apache, basis data <i>MySQL</i> dan kemampuan pemrosesan bahasa <i>PHP</i>.3. <i>Web Browser</i> yang sudah <i>Javascript Enabled</i>.

Sistem menggunakan 2 jenis *user* yang melakukan pengelolaan data yaitu *Administrator* dan *Alumni*, oleh sebab itu dalam pembuatan sistem ini, secara fisik lokasi *file-file php* dan *html* dipisahkan dalam *folder* untuk masing-masing *user* yang dapat dilihat pada gambar 5.1:



Gambar 5.1. Lokasi *folder* masing-masing *user*.

5.2 Koneksi Database

Dalam membangun sistem yang menggunakan database *MySQL* dan bahasa pemrograman web *PHP* harus ada penghubung yaitu *file php* yang menghubungkan aplikasi web dengan database di server, dengan menginisialisasikan nama server, user, password, dan nama database, yang selanjutnya akan diuji dengan menjalankan query pada server, dimana hasilnya berupa halaman kosong jika terhubung, dan pesan error “Koneksi gagal” jika gagal membangun hubungan, berikut adalah *code list* yang disimpan dalam *file connect.php*.

```
<?php
$server = "localhost";
$user = "root";
$password = "";
$database = "sebaran";
$koneksi = MySQL_connect($server,$user,$password);

if($koneksi){
    $koneksi_db = MySQL_select_db($database,$koneksi);

    if($koneksi_db){
    }else{
        print ("Koneksi gagal");
    }
}
```

```

else{
    print("Koneksi gagal");
}
?>

```

5.3 Service Google Map

Setelah ada *file connect.php* yang menghubungkan aplikasi *web* dengan *php*, barulah *developer* dapat menuliskan halaman yang memanggil *service* yang disediakan oleh *google*, adapun tahapan untuk melakukannya adalah sebagai berikut.

1. Mengambil data koordinat sebaran dari *MySQL* dengan menghasilkan keluaran XML dari *file mapxml3.php*, dengan *code list* sebagai berikut:

```

<?php

// Start XML file, create parent node
$dom = new DOMDocument("1.0");
$node = $dom->createElement("markers");
$parnode = $dom->appendChild($node);
// membuka koneksi php dan MySQL
include 'connect.php';
//query pengambilan data sebaran
$query = "select alumni.nama, jurusan.nama_jurusan,
pekerjaan.perusahaan, pekerjaan.dept, pekerjaan.job,
pekerjaan.kota, pekerjaan.sdate, pekerjaan.edate, pekerjaan.coorx,
pekerjaan.coory, alumni.id_jurusan, alumni.nim, alumni.foto,
jurusan.marker from alumni, pekerjaan, jurusan where
pekerjaan.nim=alumni.nim and alumni.id_jurusan=jurusan.id_jurusan
and pekerjaan.edate IS NULL";
$result = MySQL_query($query);

if (!$result) {
    die('Invalid query: ' . MySQL_error());
}
header("Content-type: text/xml");

// Iterasi hasil yang disimpan dalam XML
while ($row = @MySQL_fetch_array($result)){
    // ADD TO XML DOCUMENT NODE
    $node = $dom->createElement("markers");
    $newnode = $parnode->appendChild($node);
    $newnode->setAttribute("nama", $row[0]);
    $newnode->setAttribute("namap", $row[1]);
    $newnode->setAttribute("perusahaan", $row[2]);
    $newnode->setAttribute("departemen", $row[3]);
    $newnode->setAttribute("pekerjaan", $row[4]);
    $newnode->setAttribute("kota", $row[5]);
    $newnode->setAttribute("sdate", date('d F Y',
strtotime($row[6])));
    $newnode->setAttribute("edate", date('d F Y',
strtotime($row[7])));
    $newnode->setAttribute("coorx", $row[8]);
}

```

```

    $newnode->setAttribute("coory", $row[9]);
    $newnode->setAttribute("id_jurusan", $row[10]);
    $newnode->setAttribute("nim", $row[11]);
    if(empty($row[12])){
    $pp="unavailable.jpg";
    }else{
    $pp=$row[12];
    }
    $newnode->setAttribute("foto", $pp);

    if(empty($row[13])){
    $marko="none.jpg";
    }else{
    $marko=$row[13];
    }
    $newnode->setAttribute("mark", $marko);
}

echo $dom->saveXML();
}

}

?>

```

Hasil pengekseskuan *file* mapxml3.php akan menghasilkan *output node-node* XML yang berisikan data-data sebaran berserta data yang akan memunculkan *marker* pada peta. Seperti terlihat pada gambar 5.2.



Gambar 5.2. Hasil eksekusi *file* mapxml3.php.

- Menampilkan *marker* data-data sebaran dalam peta *Google Map*. *Output* XML hanya menampilkan *node-node* dalam teks, untuk menampilkan *node-node* tersebut dalam peta, maka perlu dilakukan pembuatan fungsi-fungsi dan objek peta yang menggunakan *link* ke *file* mapxml3.php untuk

mengambil data sebaran dan menampilkannya dalam *marker-marker*.

Berikut adalah *code list* yang disimpan dalam *file* *peta3.php*.

```
<?php
include 'connect.php';
?>

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html;
charset=utf-8"/>
    <title>Google Maps AJAX + MySQL/PHP Example</title>
    <script src="http://maps.google.com/maps/api/js?sensor=false"
      type="text/javascript"></script>
    <script type="text/javascript">

function load() {

var map = new google.maps.Map(document.getElementById("map"), {
  center: new google.maps.LatLng(-0.527336,114.521484),
  zoom: 5,
  mapTypeId: 'roadmap'
});
var infoWindow = new google.maps.InfoWindow;

}

// link disesuaikan dengan file php yang dibuat
downloadUrl("mapxml3.php?tahun="+jr+"&jurusan="+jb,
function(data) {
  var xml = data.responseXML;
  var markers =
xml.documentElement.getElementsByTagName("markers");
  for (var i = 0; i < markers.length; i++) {
    var name = markers[i].getAttribute("nama");
    var namap = markers[i].getAttribute("namap");
    var sdate = markers[i].getAttribute("sdate");
    var idp = markers[i].getAttribute("id_jurusan");
    var nim = markers[i].getAttribute("nim");
    var comp = markers[i].getAttribute("perusahaan");
    var dept = markers[i].getAttribute("departemen");
    var job = markers[i].getAttribute("pekerjaan");
    var mrk = markers[i].getAttribute("mark");
    var foto= markers[i].getAttribute("foto");
    var point = new google.maps.LatLng(
      parseFloat(markers[i].getAttribute("coorx")),
      parseFloat(markers[i].getAttribute("coory")));

    var html = "<table border=0 bgcolor=#000000><tr
bgcolor=#FFFFFF><td><b>" + name + "</b><hr> Alumni " + namap
+ "<br>" + job + "<br> Departemen " + dept + "<br><b>" + comp + "</b><br>sejak
"+sdate+"</td><td><img src=alumni/imgalumni/" + foto + " width=80
height=100><br><a href='profilalumni.php?nim="+nim+"'
target='blank'>Lihat Profil</a></td></tr></table>";
```



```

//      var icon;

var icon='images/marker/'+mrk;
var shadow='images/marker/shadow.png';

        var marker = new google.maps.Marker({ map: map,
position: point, icon: icon, shadow: shadow });
        bindInfoWindow(marker, map, infoWindow, html);
    }
    });
}
function bindInfoWindow(marker, map, infoWindow, html) {
    google.maps.event.addListener(marker, 'click', function() {
        infoWindow.setContent(html);
        infoWindow.open(map, marker);
        document.getElementById('#labelcari').innerHTML = 'sebaran
alumni tahun <?php echo $tahunx ?>';
    });
}

function downloadUrl(url, callback) {
    var request = window.ActiveXObject ?
        new ActiveXObject('Microsoft.XMLHTTP') :
        new XMLHttpRequest;
    request.onreadystatechange = function() {
        if (request.readyState == 4) {
            request.onreadystatechange = doNothing;
            callback(request, request.status);
        }
    };
    request.open('GET', url, true);
    request.send(null);
}
</script>
</head>
<body onLoad="load()" align="center"><br>
<center>

<table width="1000" border="0">
    <tr>
        <td width="30%">
</center>
</body>
</html>

```

Hasil eksekusi *file* peta3.php terlihat seperti pada Gambar 5.3.



Gambar 5.3. Hasil eksekusi file peta3.php.

5.4 Service SIMILE Timeline

Sama dengan pengambilan layanan *google map*, untuk menghubungkan layanan yang disediakan SIMILE dengan aplikasi *web*, file *connect.php* menghubungkan aplikasi dengan database agar layanan dapat mengolah data pada *server* untuk ditampilkan, berikut adalah tahapan untuk menampilkan data pada database *server* menjadi *event-event* timeline.

1. Menampilkan *output file json* melalui *file php* yang mengambil data melalui *query* dan menyimpannya dalam *array event*, disimpan dalam *file jsontimeline.php* seperti *code list* berikut.

```
<?php

// koneksi database
include 'connect.php';

// fungsi menampilkan atribut-atribut event

function eventAtt() {

//query pengambilan event
$eventQuery=MySQL_query("select alumni.nama, jurusan.nama_jurusan,
pekerjaan.perusahaan, pekerjaan.dept, pekerjaan.job,
pekerjaan.kota, pekerjaan.sdate, pekerjaan.edate, alumni.nim,
YEAR(alumni.eyear), alumni.id_jurusan from alumni, pekerjaan,
jurusan where alumni.nim=pekerjaan.nim and
alumni.id_jurusan=jurusan.id_jurusan") or die("Gagal mengambil
versi!");
while ($row = MySQL_fetch_array($eventQuery)) {
    $date = explode("-", $row['sdate']); // in my case, $date
is stored as yyyy-mm-dd in MySQL table.
    $phpmakedate = mktime(0, 0, 0, $date[1], $date[2],
$date[0]);
```

```

//event dengan tanggal selesai

if ($row['edate']== NULL || $row['edate'] == '0000-00-00') {
    $phpenddate = NULL;
    $durationEvent = FALSE; //JSON
    $abis='';
}
else {
    $durationEvent = TRUE; //JSON
    $enddate = explode("-", $row['edate']);
    $phpmakeenddate = mktime(0, 0, 0, $enddate[1],
    $enddate[2], $enddate[0]);
    $phpenddate = date("r",$phpmakeenddate);
    $abis='Akhir Kerja : '.$phpenddate;

}
// ----- array -----
$eventAtts[] = array (
    'start' => date("r",$phpmakedate),
    'end' => $phpenddate,
    'durationEvent' => $durationEvent,
    'description' => 'Nama : '.$row['nama'].' <br>
Perusahaan : '.$row['perusahaan'].' <br> Mulai
Kerja : '.date("r",$phpmakedate).' <br> '.$abis.'
<a href=profilalumni.php?nim='.$row['nim'].'
target=blank>Profil</a>',
    'title'=>$row['nama'].'', '.$row['perusahaan']
);
}

@MySQL_fetch_array($eventQuery);
return $eventAtts;
}

$json_data = array (
    //Atribut Timeline
    /* 'wiki-url'=>'http://simile.mit.edu/shelf',
    'wiki-section'=>'Simile Cubism Timeline',
    'dateTimeFormat'=>'Gregorian', //JSON!*/
    //Event attributes
    'events'=> eventAtt() // disini array dibentuk dari fungsi
diatas
);

$json_encoded=json_encode($json_data);
echo $json_encoded;

?>

```

Hasil eksekusi *file jsonize.php* menghasilkan *array json* berupa *event-event* sebaran alumni yang berisi atribut-atribut yang akan divisualisasikan. Hasil eksekusi *jsontimeline.php* dapat dilihat dari potongan *array list* berikut yang diambil dari eksekusi *web*.

```
{
  "events": [
    {
      "start": "Tue, 10 Jan 2012 00:00:00 +0700",
      "end": "Sat, 30 Jun 2012 00:00:00 +0800",
      "durationEvent": true,
      "description": "Nama : John Dhoe <br> Perusahaan : PT Akira Sukindo <br> Mulai Kerja : Tue, 10 Jan 2012 00:00:00 +0700 <br> Akhir Kerja : Sat, 30 Jun 2012 00:00:00 +0800 <a href=profilalumni.php?nim=0810117261444 target=blank>Profil</a>",
      "title": "John Dhoe, PT Akira Sukindo"
    },
    {
      "start": "Sun, 01 Jul 2012 00:00:00 +0800",
      "end": "Sat, 25 Aug 2012 00:00:00 +0800",
      "durationEvent": true,
      "description": "Nama : John Dhoe <br> Perusahaan : PT Trans Cooperation <br> Mulai Kerja : Sun, 01 Jul 2012 00:00:00 +0800 <br> Akhir Kerja : Sat, 25 Aug 2012 00:00:00 +0800 <a href=profilalumni.php?nim=0810117261444 target=blank>Profil</a>"
    }
  ]
}
```

- Menampilkan *event* data-data sebaran dalam timeline. *Output json array list* hanya menampilkan *list* dalam teks, untuk menampilkan *array list* tersebut dalam timeline, maka perlu dilakukan pembuatan fungsi-fungsi pemuatan *event* yang menggunakan *link* ke *file jsonized.php* untuk mengambil data sebaran dan menampilkannya dalam *event-event*. Berikut adalah *code list* yang disimpan dalam *file timelinefinal.php*.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<?php
include 'connect.php';
?>
<head>
  <title>Timeline Sebaran Alumni</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</script>
<script src="http://api.simile-widgets.org/timeline/2.3.1/timeline-api.js?bundle=true" type="text/javascript"></script>

<script type="text/javascript">

function onLoad() {

var eventSource = new Timeline.DefaultEventSource();

var bandInfos = [
/* Timeline.createBandInfo({
    eventSource: eventSource,
    width: "65%",
    intervalUnit: Timeline.DateTime.DAY,
    intervalPixels: 100
  })),*/
```

```

        Timeline.createBandInfo({
            eventSource:    eventSource,
            width:          "80%",
            intervalUnit:    Timeline.DateTime.MONTH,
            intervalPixels:  200
        }),
        Timeline.createBandInfo({
            // eventSource:    eventSource,
            // date:          focusDate, //"Jun 28 2009 00:00:00 GMT",
            width:          "20%",
            intervalUnit:    Timeline.DateTime.YEAR,
            intervalPixels:  100
        })
    ];

    bandInfos[1].syncWith = 0;
    bandInfos[1].highlight = true;

    var ling="jsonized.php?";
    jr="<?php echo $tahunx; ?>";
    jb="<?php echo $jurusanx; ?>";
    tl      =      Timeline.create(document.getElementById("xxx"),
bandInfos);
    tl.loadJSON(ling+''+(new    Date())+'&tahun='+jr+'&jurusan='+jb,
function(json, url) {
    eventSource.loadJSON(json, url);
});
}
var resizeTimerID = null;
function onResize() {
    if (resizeTimerID == null) {
        resizeTimerID = window.setTimeout(function() {
            resizeTimerID = null;
            tl.layout();
        }, 500);
    }
}

</script>

</head>
<body onload="onLoad();" onresize="onResize();">

</center>

</body>
</html>

```

5.5 Pengujian

Pengujian dilakukan dengan menjalankan fungsi pada masing-masing *user*, *user Administrator* mengelola data jurusan dan kota sebaran, *user Jurusan* mengelola data alumni setiap tahun lulusan, dan *user Alumni* mengelola profil dan sebarannya sendiri.

5.5.1 User Administrator

User Administrator berfungsi sebagai pengelola data Jurusan dan kota sebaran.

1. Login Administrator



Gambar 5.4. Tampilan form *Login Administrator*.

Agar dapat masuk kedalam sistem, *administrator* harus *Login* terlebih dahulu dengan mengisi *username* dan *password* dalam form *Login Administrator*. Seperti yang terlihat pada gambar 5.4.

2. Pengelolaan Jurusan : Daftar Jurusan



Gambar 5.5. Tampilan Data Jurusan.

Seperti yang terlihat pada gambar 5.5 diatas, daftar Jurusan melihatkan jurusan yang ada di UIN SUSKA RIAU, saat ini ada 5 jurusan yaitu Matematika Terapan, Teknik Informatika, Sistem Informasi, Teknik Elektro, dan Teknik Industri.

Tabel 5.2 ID Jurusan dan Nama Jurusan

ID Jurusan	Nama Jurusan
jur01	Teknik Informatika
jur02	Sistem Informasi
jur03	Teknik Elektro
jur04	Matematika Terapan
jur05	Teknik Industri

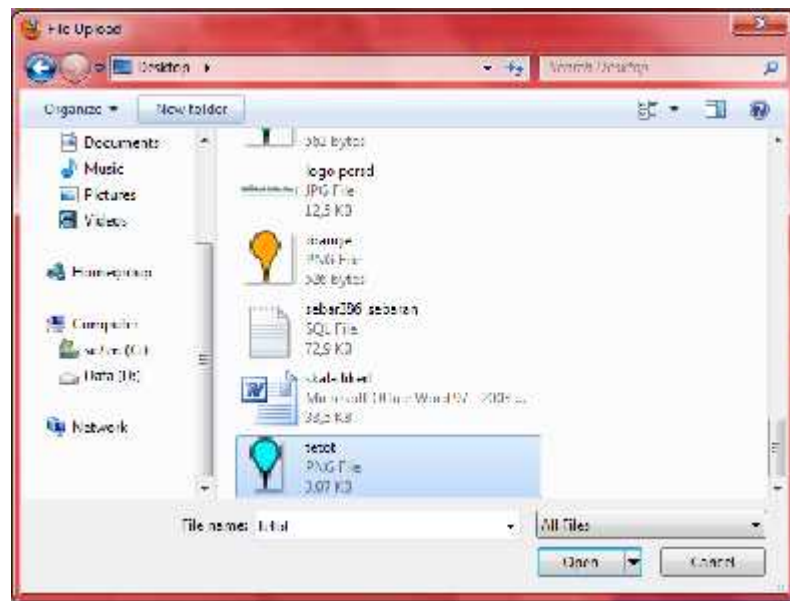
- Kode : secara *default* sudah berisi 4 digit pertama 1011, untuk *input* yang *valid* admin harus mengisi 4 digit terakhir, kode jurusan dan kode jurusan, dimana dalam contoh, jurusan Teknik Elektronika Telekomunikasi memiliki kode jurusan 74 dan kode Jurusan 21.
- Nama Jurusan : Tidak boleh kosong ataupun terlalu pendek (5 karakter).
- Diploma : memilih jenis Program Studi, D3 untuk 3 tahun dan D4 untuk 4 tahun.
- Inisial : Inisial Jurusan yang terdiri maksiman 3 karakter.
- Tanggal Berdiri : diambil dari *date picker* dengan menekan tombol *icon* kalender yang ada disamping field Tanggal berdiri.

3. Perubahan Data Jurusan

Gambar 5.6. Form perubahan data jurusan.

Data jurusan yang sudah ditamah belum memiliki *marker* yang nanti dibutuhkan dalam melakukan visualisasi, jadi admin harus menambahkan sebuah *marker* untuk jurusan. Untuk menambahkan tekan tombol *Browse*, setelah mendapatkan *file markernya*, tekan *Upload Marker* untuk menyimpan *marker*. Seperti terlihat pada gambar 5.7.

4. Penambahan *Marker*.



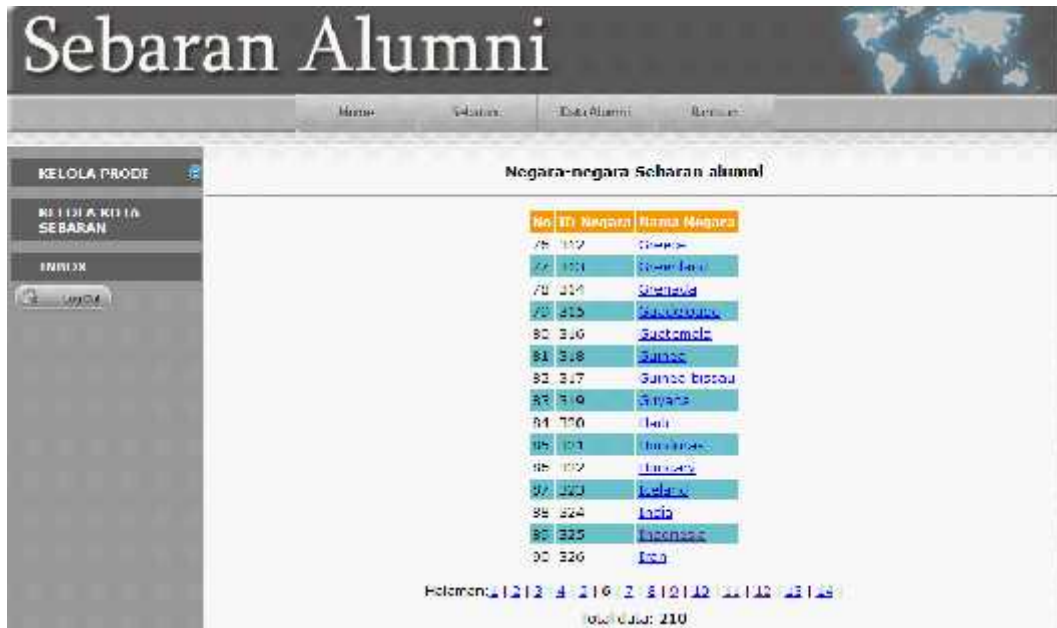
Gambar 5.7. Pengambilan *marker* dari *folder*.

Marker adalah *file image* yang secara visual dijadikan penanda dalam visualisasi pada peta, format *image* yang *valid* adalah format .png yang memiliki *background* transparan. Hasil dari perubahan *marker* dapat dilihat dalam form perubahan data Jurusan dimana yang sebelumnya *marker* sementara yang masih hitam sekarang menjadi warna yang sesuai dengan *file* yang di unggah tadi. Seperti yang terlihat pada gambar 5.8.



Gambar 5.8. Hasil perubahan *marker*.

5. Pengelolaan Kota Sebaran : Negara Sebaran



The screenshot shows the 'Sebaran Alumni' web application interface. The header includes the title 'Sebaran Alumni' and navigation links: 'Home', 'Sebaran', 'Data Alumni', and 'Alumni'. The left sidebar contains 'KELOLA PRODE', 'INDUK KOTA SEBARAN', and 'INDEX'. The main content area is titled 'Negara-negara Sebaran alumni' and displays a table with columns 'No', 'ID Negara', and 'Nama Negara'. The table lists 16 countries with their respective IDs and names, each with a link to a city list. Below the table, there are pagination controls showing 'Halaman: 1 2 3 4 5 6 7 8 9 10 11 12 13 14' and 'Total data: 210'.

No	ID Negara	Nama Negara
76	112	Ghana
77	113	Guinea
78	114	Guinea
79	115	Guinea
80	116	Guinea
81	118	Guinea
82	117	Guinea
83	119	Guinea
84	120	Guinea
85	121	Guinea
86	122	Guinea
87	123	Guinea
88	124	Guinea
89	125	Guinea
90	126	Guinea

Gambar 5.9. Daftar Negara-negara Sebaran Alumni.

Walaupun data yang diperlukan untuk sebaran alumni adalah kota, namun untuk mengelola data-data kota tersebut, diperlukan mekanisme penyusunan kota-kota dengan mengelompokkan berdasarkan negara, seperti terlihat pada Gambar 5.9, nama-nama yang tampil merupakan *link* yang akan mengarahkan ke daftar kota, dengan contoh menekan *link* negara Indonesia

5.5.2 User Alumni

1. Login User Alumni



Gambar 5.10 Tampilan halaman sebaran.

Berdasarkan gambar 5.10 diatas, pada halaman sebaran ini terdapat tombol “Saya alumni UIN SUSKA RIAU!”, tombol tersebut merupakan tombol *Login* untuk alumni, dimana pada saat menekan tombol tersebut akan muncul *pop up* form *Login*

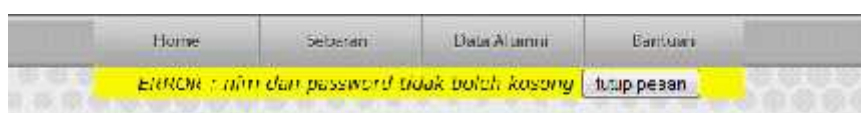


Gambar 5.11 Tombol untuk *Login*.

Untuk masuk ke sistem, setelah menekan tombol “Saya alumni UIN SUSKA RIAU!”, maka alumni harus mengisi NIM Alumni dan *Password* yang sudah dimasukkan oleh Jurusan, pesan tidak *valid* akan muncul jika alumni belum mengisi data *Login* dengan benar, tidak boleh kosong, NIM harus dikenal, dan *password* tidak boleh salah.



Gambar 5.12 *Pop up Login Form*.



Gambar 5.13 Pesan *Error Nim dan Password* yang kosong.



Gambar 5.14 Pesan *Error Nim* yang tidak dikenal.

2. Melihat Profil

Profil sendiri dapat dilihat dengan menekan Profil Saya pada Menu *Sidebar*. Seperti yang terlihat pada gambar 5.15:



Gambar 5.15 Profil Alumni

3. Mengubah Profil

Profil sendiri dapat dilihat dengan menekan Edit Profil pada Menu *Sidebar*. Seperti yang terlihat pada gambar 5.16:



Gambar 5.16 Form Perubahan Data Profil.

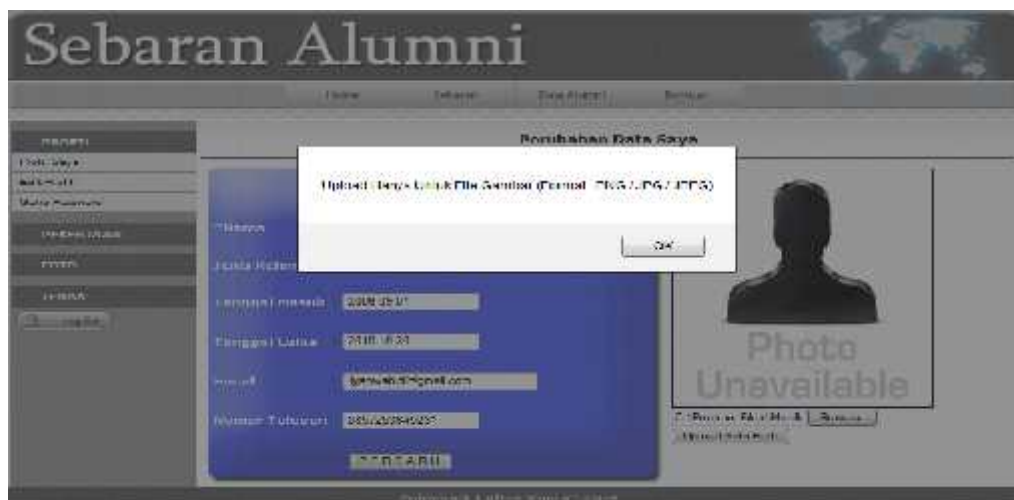
Perubahan datap profil harus berisikan data email yang *valid* (contoh: *username@domain.com*), nomor telepon yang *valid* (dalam bentuk angka), dan nama alumni, nomor telepon, serta alamat email tidak boleh kosong.



Gambar 5.17 Fom Perubahan Data Profil.

4. Mengganti foto profil.

Foto secara *default* akan menampilkan gambar “*photo unavailable*”, sehingga alumni harus melakukan *upload* foto, *validasi* ada untuk format *file* yang akan di *upload* yaitu PNG, JPG, atau JPEG. Seperti yang terlihat pada gambar 5.18:



Gambar 5.18 *Validasi* Pergantian Foto Profil.

5. Pergantian *password*.

Secara *default* alumni sudah memiliki *password*, namun jika alumni ingin merubah *password*, maka dengan menekan tombol ganti *password* pada menu *side bar* alumni dapat mengisi *password* baru dengan kriteria tidak kurang dari 6 karakter dan tidak lebih dari 8 karakter. Seperti yang terlihat pada gambar 5.19:

The image shows a web form titled "Form Pergantian Password". It has a blue header bar. Below the header, there are two input fields. The first is labeled "Password Baru" and contains six dots. The second is labeled "Ketik Ulang Passowrd Baru :" and is empty. Below these fields is a button labeled "Ganti Password".

Gambar 5.19 Form pergantian *password*.

6. Menambah Pekerjaan

Untuk melihat pekerjaan terkini, alumni dapat menekan Pekerjaan saya pada menu *side bar*, jika data pekerjaan belum ada, maka hasilnya akan terlihat seperti Gambar 5.20:



Gambar 5.20 Data Pekerjaan terkini yang belum ada.

7. Penambahan Pekerjaan.

Untuk menambah data pekerjaan, alumni dapat menekan tombol tambah pekerjaan dibawah hasil pekerjaan terkini yang kosong, pekerjaan hanya bisa ditambah jika belum ada pekerjaan terkini yang tampil, untuk menambah pekerjaan lagi, pekerjaan sebelumnya harus diakhiri. Seperti yang terlihat pada gambar 5.21:

Tambah Sebaran Baru (Informasi Lokasi dan Masa Pekerjaan)

Seret marker merah untuk mengatur posisi sebaran

Perusahaan:

Departemen:

Jabatan/Pekerjaan:

Kota:

Tanggal Mulai:

Tanggal Selesai:

Koordinat X:

Koordinat Y:

Tambahkan Informasi Sebaran

Gambar 5.21 Form penambahan pekerjaan.

8. Mengakhiri Pekerjaan

Pekerjaan Terkini

Perusahaan : Riau Televisi

Departemen : Broadcasting

Jabatan/Pekerjaan : Editor

Kota : Pekanbaru

Mulai Kerja : 06 Desember 2010

Akhiri Pekerjaan

Gambar 5.22 Pekerjaan terkini.

Seperti yang terlihat pada gambar 5.22 diatas, untuk mengakhiri suatu pekerjaan alumni dapat menekan tombol merah “Akhiri Pekerjaan” pada form selanjutnya alumni bisa memasukkan tanggal berakhirnya pekerjaan tersebut. Dapat dilihat pada gambar 5.23:

Selesaikan Pekerjaan

Perusahaan : Riau Televisi

Departemen : Broadcasting

Jabatan/Pekerjaan : Editor

Kota : Pekanbaru

Tanggal Mulai : 06 Desember 2010

Tanggal Selesai :

Akhir Masa

Gambar 5.33 Form penyelesaian pekerjaan.

Hasil berakhirnya pekerjaan dapat dilihat di tabel pekerjaan yang sudah selesai dimana pekerjaan sebelumnya tidak ada pada tabel pekerjaan terkini, karena pada bagian pekerjaan terkini sudah tertulis “Pekerjaan Terkini Belum Ada” dan Pekerjaan yang sudah diakhiri telah berada di tabel “Pekerjaan yang sudah selesai”. Seperti yang terlihat pada gambar 5.24:

Pekerjaan Terkini								
<p style="text-align: center;">Pekerjaan Terkini Belum Ada</p>  <p style="text-align: center;">Tambah Pekerjaan</p>								
Pekerjaan yang sudah selesai								
Masa Kerja Anda Pada Pekerjaan Sebelumnya Sudah Berakhir								
No	Perusahaan	Departemen	Pekerjaan	Kota	Tanggal Mulai	Tanggal Selesai	Ubah	Hapus
1	Riau Televisi	Broadcasting	Editor	Pekanbaru	06 Desember 2010	27 May 2011		
2	Metro TV	Broadcasting	Editor	Jakarta	06 June 2011	23 July 2012		

Gambar 5.24 Tabel Pekerjaan terkini dan Pekerjaan yang sudah selesai.

9. Mengubah informasi sebaran.

Seperti yang terlihat pada gambar 5.25:

Ubah Informasi Sebaran	
<p>Search marker on map to manipulate position sebaran!</p> 	
*Perusahaan:	Metro TV
*Departemen:	Broadcasting
*Pekerjaan/Jabatan:	Editor
*Kota:	Jakarta
*Tanggal Mulai:	2011/06/06
Tanggal Selesai:	2012/07/23
Koordinat X:	-6.195100
Koordinat Y:	106.734434
<input type="button" value="Ubah Informasi Sebaran"/>	

Gambar 5.25 Form perubahan informasi sebaran.

a. *Hasil*

1. Hasil visualisasi dalam peta.

Seperti yang terlihat pada gambar 5.26:



Gambar 5.26 Hasil Sebaran dalam peta

2. Hasil zoom dalam peta.

Sebaran akan tampak hanya jika alumni memiliki data pekerjaan terkini, dari peta dapat dilihat keberadaan alumni dimana jurusan alumni tersebut dapat dilihat pada menu pencarian yang menunjukkan masing-masing warna untuk jurusan yang ada.



Gambar 5.27 Hasil Zoom in Peta.

Dalam peta dalam keadaan zoom yang jauh ada kemungkinan *marker* yang tampak hanyalah satu, untuk memastikan, *user* dapat melakukan zoom in untuk melihat gambar yang lebih *details*, walaupun bukan dalam satu koordinat bisa saja dua atau beberapa *marker* alumni terlihat saling menimpa, dengan melakukan zoom in kita bisa melihat *marker* keberadaan alumni lebih jelas. Seperti yang terlihat pada gambar 5.27 diatas.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Setelah melakukan pengujian dengan menjalankan fungsi *user* pada sistem, maka dapat disimpulkan bahwa:

1. Dengan adanya integrasi aplikasi Pengembangan Visualisasi Universitas Islam Negeri Sultan Syarif Kasim Riau berbasis android dengan layanan yang diberikan oleh *Google*, visualisasi berbasis lokasi dan waktu dapat dibangun beserta informasi-infromasi yang terkait.
2. Alumni dapat melakukan *update* informasi sebarannya yaitu tempat kerja, lokasi, dan tanggal mulai mereka bekerja.
3. Jurusan mendapatkan manfaat dari sistem yang menyimpan data sebaran, sehingga jurusan dapat memantau keberadaan alumni terkini, dengan demikian dapat dikatakan sistem membantu Jurusan dalam melakukan *tracer study*.
4. Berdasarkan analisa kuisisioner manfaat yang direspon paling baik dari masing-masing *user* adalah, kemampuan sistem menyajikan data alumni untuk *user* publik, kemudahan pengelolaan data alumni, dan kemampuan sistem melakukan visualisasi sebaran untuk alumni.

6.2 Saran

Sistem sebaran berbasis android ini dapat berinteraksi dengan perangkat mobile yang dibangun dengan *client application* khusus untuk alumni sehingga bagi alumni yang mobilitasnya tinggi dapat memperbarui informasi sebarannya dimana saja

DAFTAR PUSTAKA

- Amri, Syaiful.(2011). *Membangun Sistem Navigasi di Surabaya Menggunakan Google Maps API*. Diambil 5 November 2012 dari : <http://repo.eepis-its.edu/1220/1/paper.pdf>
- Bourdon, Romain.(t.t). *Start with Wampserver*. Diambil 17 Januari 2013 dari: <http://www.wampserver.com/en/>
- Croxall, Brian.(2001). *Timeline Tutorial*. Diambil 14 Desember 2012 dari: <http://briancroxall.net/TimeineTutorial/>
- Dabernig, Josef.(2008). *Creating interactive web pages using the Exhibit framework*. Diambil 20 Desember 2012 dari : <http://simile-widgets.googlecode.com/files/Creating%20interactive%20web%20pages%20using%20the%20Exhibit%20framework.pdf>
- Dabernig, Josef.(2009). *Timeline Event Display*. Diambil 16 Januari 2013 dari : <http://code.google.com/p/simile-widgets/wiki/>
- Goldsky.(2010).*Timeline load event dynamically*. Diambil 18 Januari 2013 dari : http://www.simile-widgets.org/wiki/Timeline_LoadEventsDynamically
- Google Geo Team. (2008). *Creating a Store Locator with PHP, MySQL & Google Maps*. Diambil 3 Desember 2011 dari : <http://code.google.com/apis/maps/articles/phpsqlsearch.html>.
- Google Geo Team. (2008). *Google Maps JavaScript API v3*. Diambil 3 Desember 2012 dari : <https://developers.google.com/maps/documentation/javascript/tutorial>
- Horton, Sarah & Lynch, Petrick J.*SITE DESIGN : Creating site diagrams*. (2004). Diambil 24 Februari dari : <http://www.webstyleguide.com/wsg2/site/>
- Kadir, Abdul. (2009). *Membuat aplikasi web dengan PHP dan database MySQL*. Yogyakarta : Penerbit ANDI.
- Madcoms. (2009). *Membangun Website Profesional dengan Adobe CS4, PHP, & MySQL*. Yogyakarta: Andi.
- Prasetyo, Didik Dwi. (2003). *TIPS dan TRIK Kolaborasi PHP dan MySQL untuk Membuat Web Database yang Interaktif*. Yogyakarta : Penerbit ANDI.
- Sidik, Bertha (2003). *MySQL*. Bandung : Informatika Bandung
- Simamora, B. (2005). *Analisis Multivariat Pemasaran*. Jakarta : Gramedia.